

Saltés

PROGRAMAS EXPLICADOS DE

BASIC JUNIOR

RICARDO AGUADO-MUÑOZ
AGUSTIN BLANCO
ENRIQUE RUBIALES
JAVIER ZABALA
RICARDO ZAMARREÑO

De los mismos autores:

BASIC BASICO
CURSO DE PROGRAMACION

Es un libro de texto o divulgativo, fruto de la experiencia didáctica de tres años en el Instituto Piloto «Cardenal Herrera Oria».

Explica la programación en BASIC. Muchos de los ejercicios que propone se resuelven en el libro Programas Comentados de BASIC BASICO.

PROGRAMAS COMENTADOS
DE BASIC BASICO

Ciento ocho problemas de programación explicados y comentados.

Cincuenta de ellos responden a los ejercicios propuestos en el libro BASIC BASICO CURSO DE PROGRAMACION.

Contiene:

Juegos. Simulaciones. Gráficos y color. Cálculo numérico.

BASIC JUNIOR
INICIACION A LA PROGRAMACION

Está destinado a escolares y principiantes.

Es formativo y ameno. Tiene numerosos ejemplos divertidos.

BASIC JUNIOR te invita a jugar el apasionante juego de la programación de ordenadores.

PROGRAMAS EXPLICADOS DE

BASIC JUNIOR

RICARDO AGUADO-MUÑOZ
AGUSTIN BLANCO
ENRIQUE RUBIALES
JAVIER ZABALA
RICARDO ZAMARREÑO

DISTRIBUYE:
GRUPO DISTRIBUIDOR EDITORIAL, S. A.
Don Ramón de la Cruz, 67
28001 MADRID
Tfno.: 91-401 12 00

Edita: Computer School. C/. Brístol, 10. 28028 Madrid

I.S.B.N.: 84-86529-00-X
Depósito legal: M. 27.612-1986
Imprime: Hijos de E. Minuesa, S. L. Ronda de Toledo, 24. 28005 Madrid

INDICE

PRESENTACION

La programación de ordenadores, que hasta hace poco estaba limitada a un núcleo pequeño de especialistas, se ha convertido en un fenómeno importantísimo de aprendizaje y práctica, que afecta a miles de personas: es *la nueva alfabetización.*

Este libro está destinado a todos los jóvenes que quieran iniciarse en *el arte de programar ordenadores,* bien sea en su propia casa, o en el colegio. Ofrecemos al joven lector *cien programas explicados y comentados,* que son los cien ejercicios propuestos en nuestra obra anterior *BASIC JUNIOR Iniciación a la programación.* Ambos libros son complementarios y, al mismo tiempo, independientes.

Todos los programas están preparados para que puedan funcionar sin dificultad en cualquier ordenador personal. Naturalmente, el lector puede y debe adecuarlos a su propio ordenador. Para los usuarios de *Spectrum* y *Commodore,* nosotros mismos indicamos a pie de página las escasas adaptaciones que es preciso realizar.

Mucha gente utiliza los ordenadores únicamente para jugar. Nosotros estamos convencidos de que, hoy por hoy, *el verdadero juego es programar;* porque los actos de inteligencia necesarios para desmenuzar un problema, comprenderlo y luego contárselo al ordenador, son lo suficientemente interesantes como para cautivar la voluntad de la mayoría de las personas, sobre todo de las más jóvenes.

Cada enunciado de este libro es un reto a tu inteligencia. Programa la solución a tu manera y luego compárala con el listado que te ofrecemos. Muchas veces darás con soluciones distintas y, a menudo, mejores que las nuestras. Así comprenderás que la programación de ordenadores no despersonaliza, sino que fomenta la iniciativa y la creatividad.

Una vez más tenemos que agradecer a nuestro amigo y compañero Saltés el diseño de la portada y las ilustraciones interiores. Sin ellas este libro perdería muchísimo.

Con esta obra completamos la colección BASIC BASICO-BASIC JUNIOR, con la que hemos pretendido facilitar el estudio del BASIC en dos niveles de profundidad. Confiamos en que este libro sea un digno miembro de la familia y que sea acogido por el público con la misma simpatía y benevolencia que sus hermanos mayores.

LOS AUTORES

1. DIBUJO DE UN PEZ (1.1)

Prepara un programa que dibuje este pez:

```
      * * * *
    * *     *   *
  *   *       * *
    * *     *   *
      * * * *
```

Ya sabes que el ordenador dibuja o imprime de línea en línea. Como este pez tiene cinco líneas, puedes escribir el siguiente programa:

```
(1) 10 PRINT "      * * * *"
    20 PRINT "    * *     *   *"
    30 PRINT "*   *       * *"
    40 PRINT "    * *     *   *"
    50 PRINT "      * * * *"
(2) 60 END
```

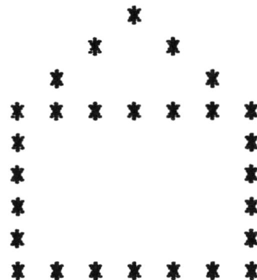
No olvides dejar los espacios en blanco necesarios. Se consiguen dando a la tecla (SPACE) o a la barra espaciadora, según el ordenador que tengas.

(1) Busca en tu ordenador la tecla del asterisco (*). En algunos ordenadores está entre las teclas de operaciones. En los Sinclair se obtiene pulsando a la vez la B y (SYMBOL SHIFT)

(2) En los ordenadores Sinclair poner STOP.

2. DIBUJO DE UNA CASA (1.2)

Prepara un programa que dibuje la casa:



Te proponemos la siguiente solución:

```
10 PRINT "      *"
20 PRINT "    *   *"
30 PRINT "  *       *"
40 PRINT "* * * * * * *"
50 PRINT "*           *"
60 PRINT "*           *"
70 PRINT "*           *"
80 PRINT "*           *"
90 PRINT "* * * * * * *"
100 END
```

En el listado del programa puedes ver la casa ya dibujada entre las comillas de las líneas PRINT. Así resulta muy fácil programar dibujos; se ponen instrucciones PRINT y el dibujo entre las comillas, dejando los espacios en blanco necesarios.

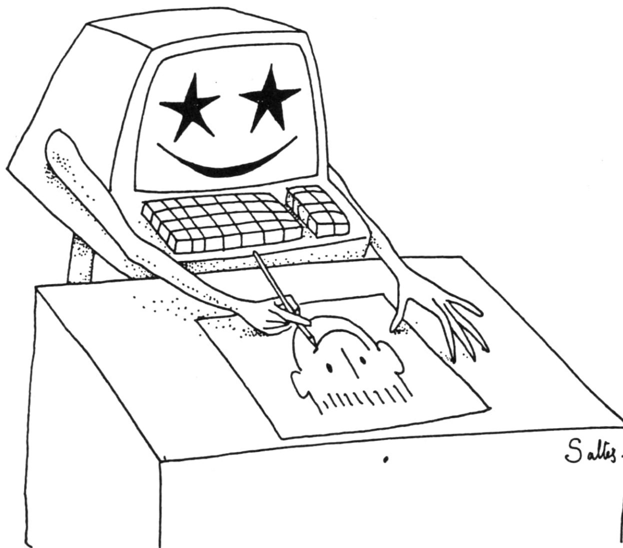
3. DIBUJO DE UN SEÑOR (1.3)

Haz un programa que dibuje la cara de este señor:

```
/ / / / / /
!         !
! ( . . ) !
!         !
!   , ,   !
!   - -   !
 V V V V V
```

Es igual de fácil que los anteriores, pero ahora se utilizan varias teclas / ! (.) , - V y no únicamente el asterisco.

```
100 PRINT "/ / / / / /"
110 PRINT "!         !"
120 PRINT "! ( . . ) !"
130 PRINT "!         !"
140 PRINT "!   , ,   !"
150 PRINT "!   - -   !"
160 PRINT " V V V V V"
170 END
```



4. VARIOS DIBUJOS (1.4)

Haz cuatro programas para dibujar las figuras:

```
* * * * *     *             * * * * *     * * * * *
*       *     * *             * * * *     * * * * *
*       *     * * *             * * *     * * * * *
*       *     * * * *             * *     * * * * *
* * * * *     * * * * *             *     * * * * *
```

En cada programa tienes que utilizar 5 instruciones PRINT (no te olvides de las comillas, ni de incluir los espacios que sean necesarios para dibujar las figuras).

```
10 PRINT "* * * * *"
20 PRINT "*       *"
30 PRINT "*       *"
40 PRINT "*       *"
50 PRINT "* * * * *"
60 END
```

```
10 PRINT "*"
20 PRINT "* *"
30 PRINT "* * *"
40 PRINT "* * * *"
50 PRINT "* * * * *"
60 END
```

```
10 PRINT "* * * * *"
20 PRINT "  * * * *"
30 PRINT "    * * *"
40 PRINT "      * *"
50 PRINT "        *"
60 END
```

```
10 PRINT "* * * * *"
20 PRINT "* * * * *"
30 PRINT "* * * * *"
40 PRINT "* * * * *"
50 PRINT "* * * * *"
60 END
```

5. LAS COMILLAS (2.1)

Intenta predecir lo que tiene que salir en la pantalla con estos programas. Pásalos por tu "micro" y compara los resultados con tus predicciones:

```
10 LET A=5
20 PRINT "A"
30 PRINT A
40 END
```

```
10 LET A=3
20 LET B=2
30 PRINT "A+B="
40 PRINT A+B
50 END
```

```
10 PRINT 3+7
20 PRINT "3+7"
30 END
```

Cada programa tiene dos instrucciones PRINT: una con comillas y otra sin comillas.

Los PRINT con comillas "*dibujan*" lo que tienen dentro, es decir: A, A + B = y, por último, 3 + 7.

En cambio, cuando no hay comillas, la instrucción PRINT hace los cálculos; por ejemplo: 3 + 7 = 10. Si se han utilizado variables (A ó A + B), PRINT hace los cálculos con el contenido (el valor) de esas variables. Tras calcular el resultado, lo imprime: 5, 5 y 10.

Lo que resulta en la pantalla es, respectivamente:

```
A
 5
READY.
```

```
A+B=
 5
READY.
```

```
 10
3+7
READY.
```

6. DE UNO EN UNO (2.2)

¿Qué hace el ordenador con este programa?

```
10 LET N=0
20 LET N=N+1
30 PRINT N
40 LET N=N+1
50 PRINT N
60 LET N=N+1
70 PRINT N
80 END
```

En las instrucciones LET el ordenador calcula el valor de N.

Línea N.º	Valor de N
10	N = 0
20	N = 0 + 1 = 1
40	N = 1 + 1 = 2
60	N = 2 + 1 = 3

Las instrucciones PRINT, que están en las líneas 30, 50 y 70, imprimen los respectivos valores de N. En la pantalla saldrá:

```
RUN
 1
 2
 3
READY.
```

7. LONGITUD DE CIRCUNFERENCIAS (2.3)

Prepara un programa para que el ordenador te pida (INPUT) el radio R de una circunferencia y a continuación escriba (PRINT) la longitud L (L = = 2 × 3,14 × R).

Vamos a ir paso a paso, como en el enunciado:

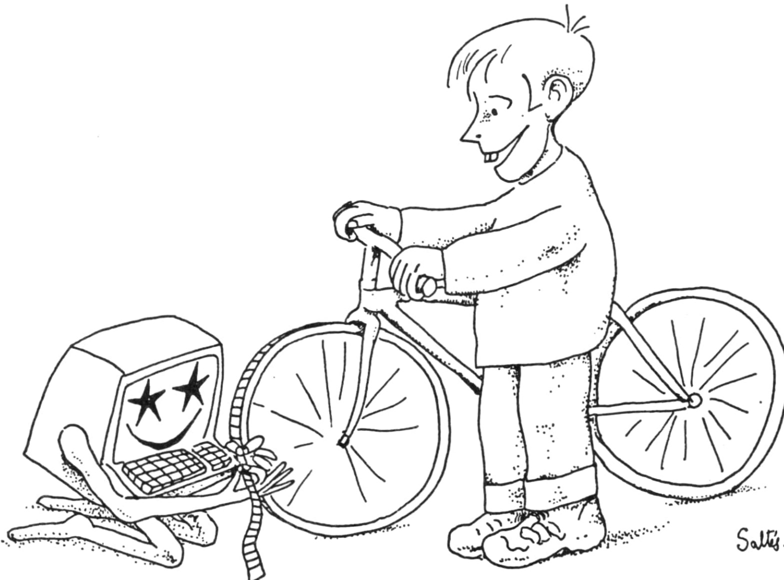
— Primero preguntamos el radio.
— Después calculamos la longitud de la circunferencia.
— Por último, imprimimos el resultado.

Mira el programa:

```
10 INPUT R
20 LET L=2*3.14*R
30 PRINT L
40 END
```

También se pueden juntar las líneas 20 y 30 en una sola:

```
10 INPUT R
20 PRINT 2*3.14*R
30 END
```



8. MICRODESCUENTO (2.4)

En el micromercado hacen un microdescuento de un 3,5 % sobre el valor de la compra efectuada. Haz un programa para que el ordenador tome (INPUT) el precio normal y escriba (PRINT) el precio reducido.

Como puedes ver, hay cuatro pasos:

— *Preguntar* el Precio Normal (INPUT).
— *Calcular* el Descuento (LET).
— *Calcular* el Precio Reducido, que es la diferencia (LET).
— *Imprimir* el Precio Reducido (PRINT).

La variable PN será el Precio Normal, D será el Descuento y PR el Precio Reducido.

```
10 INPUT PN
20 LET D=PN*3.5/100
30 LET PR=PN-D
40 PRINT PR
50 END
```

Para mucha gente resulta más cómodo calcular directamente el Precio Reducido. Se obtiene desarrollando la fórmula de la línea 30:

```
PR=PN-D=PN-PN*3.5/100=PN*(1-3.5/100)
```

Operando resulta:

```
PR = PN*0.965
```

El programa queda ahora más corto:

```
10 INPUT PN
20 LET PR=PN*.965
30 PRINT PR
40 END
```

9. REFRANES (3.1)

Seguro que conoces estos refranes:

A1$	QUIEN A BUEN ARBOL SE ARRIMA
A2$	BUENA SOMBRA LE COBIJA
B1$	DIME CON QUIEN ANDAS
B2$	Y TE DIRE QUIEN ERES
C1$	CRIA CUERVOS
C2$	Y TE SACARAN LOS OJOS

Haz un programa que asigne a las variables de cadena la frase correspondiente, y que luego escriba los refranes completos. También puedes mezclar una parte de un refrán con otra de otro y construir unos refranes muy originales.

```
10 LET A1$="QUIEN A BUEN ARBOL SE ARRIMA"
20 LET A2$="BUENA SOMBRA LE COBIJA"
30 LET B1$="DIME CON QUIEN ANDAS"
40 LET B2$="Y TE DIRE QUIEN ERES"
50 LET C1$="CRIA CUERVOS"
60 LET C2$="Y TE SACARAN LOS OJOS"
70 PRINT A1$+" "+A2$
80 PRINT B1$+" "+B2$
90 PRINT C1$+" "+C2$
100 END
```

Como sabes, los refranes han de ponerse entre comillas en las instrucciones LET, por ser cadenas de caracteres.

En las instrucciones PRINT hemos puesto también comillas, pero esta vez contienen entre ellas un espacio en blanco. Es necesario para que la última palabra de A1$ quede separada de la primera de A2$, y lo mismo con los otros dos refranes. Si no, si se quitan los espacios, saldrían pegadas las dos partes de cada refrán:

...SE ARRIMABUENA SOMBRA...
...QUIEN ANDASY TE DIRE...
...CRIA CUERVOSY TE SACARAN...

Para estropear los refranes, como dice la última parte del enunciado, puedes modificar las líneas finales poniendo:

```
70 PRINT A1$+"  "+C1$
80 PRINT B2$+"  "+C2$
90 PRINT B1$+"  "+A2$
```



10. TE QUIERO (3.2)

Haz un programa que, mediante INPUT, guarde en la variable N$ el nombre de una persona; que asigne a F$ la cadena "TE QUIERO" y que, por último, escriba N$ + F$.

Para pedir el nombre hace falta la instrucción INPUT N$. Luego se guarda (LET) la expresión "TE QUIERO" en la variable F$. Al escribir N$ + F$ tienes que intercalar un espacio en blanco entre las dos, para que no salgan juntas, igual que en el ejercicio anterior.

```
10 INPUT N$
20 LET F$="TE QUIERO"
30 PRINT N$+" "+F$
40 END
```

Si el nombre es ANTONIO, por ejemplo, en la pantalla saldrá:

ANTONIO TE QUIERO

Mientras que si suprimes las comillas y el espacio en blanco de la línea 30, quedaría junto:

ANTONI<u>OT</u>E QUIERO

11. TRABALENGUAS (3.3)

Conocerás el trabalenguas que dice:

El cura de Alcañiz,
a las narices llama nariz
y el cura de Alcañices
a la nariz llama narices.

Fíjate que los trozos subrayados se repiten. Puedes asignar cada uno de ellos a una variable de cadena y hacer un programa que escriba el trabalenguas como suma de variables.

```
10 LET A$="EL CURA DE ALCAÑI"
20 LET B$="A LA"
30 LET C$="NARI"
40 LET D$="CES"
50 LET E$="LLAMA NARI"
60 PRINT A$+"Z"
70 PRINT B$+"S "+C$+D$+" "+E$+"Z"
80 PRINT "Y "+A$+D$
90 PRINT B$+" "+C$+"Z "+E$+D$
100 END
```

Cada trozo de verso que está subrayado se guarda en una variable de cadena. Para esto sólo empleamos cinco líneas, porque los trozos subrayados se repiten.

Después se imprimen los cuatro versos mediante las líneas 60 a 90. Cada verso resulta de juntar, mediante la suma, varios trozos: tanto los que están guardados en variables como otros que ponemos entre comillas.

El reparto de los espacios en blanco para que las palabras no queden pegadas, resulta más variado en este programa.

12. VENECIA ENGONDOLADA (3.4)

Con las mismas ideas que en el ejercicio anterior, puedes programar este trabalenguas:

La ciudad de Venecia está engondolada,
quién la desengondolará;
el desengondolador que la desengondole
buen desengondolador será.



En este trabalenguas sólo se repite lo que se refiere a las góndolas. Por eso utilizaremos únicamente dos variables de cadena:

A$ = "ENGONDOL"
B$ = " DES" (con un espacio al principio)

El resto de cada verso habrá que ponerlo entre comillas:

```
10 LET A$="ENGONDOL"
20 LET B$=" DES"
30 PRINT "LA CIUDAD DE VENECIA ESTA "+A$+"ADA"
40 PRINT "QUIEN LA"+B$+A$+"ARA"
50 PRINT "EL"+B$+A$+"ADOR QUE LA"+B$+A$+"E"
60 PRINT "BUEN"+B$+A$+"ADOR SERA"
70 END
```

Si te cuesta entender alguna cosa, mira los tres programas anteriores, pues en ellos se utilizan y se explican los mismos trucos empleados aquí.

13. AREA DEL TRIANGULO (4.1)

Prepara un programa para que el ordenador pida (INPUT) la base B y la altura A de un triángulo y que luego escriba (PRINT) el área S.

$$S = \frac{1}{2} B \times A$$

La fórmula del área del triángulo se escribe en BASIC así:

$$S = 1/2 * B * A \qquad (1)$$

O también:

$$S = B * A/2$$

La multiplicación y la división son dos operaciones del mismo nivel, por lo que en ambas expresiones se efectúan las operaciones de izquierda a derecha. Por ejemplo, en la fórmula (1), la primera operación que hace la máquina es el cociente, y queda:

$$S = 0.5 * B * A$$

Así se ve claramente que la fórmula (1) en BASIC, responde al área del triángulo escrita arriba.

```
10 INPUT B
20 INPUT A
30 LET S=1/2*B*A
40 PRINT S
50 END
```

14. VELOCIDAD (4.2)

Un corredor hace los 100 metros en 9,8 segundos. Haz un programa para obtener la velocidad en kilómetros/hora:

$$V = \frac{100 \times 3600}{9{,}8 \times 1000}$$

Para pasar a BASIC esta fórmula, tenemos que poner entre paréntesis el denominador:

```
V = 100*3600/(9.8*1000)
```

El programa puede quedar así:

```
10 LET V=100*3600/(9.8*1000)
20 PRINT V
30 END
```

Si no se pone el paréntesis, el micro se cree que el número 1000 está en el numerador. Mira cómo se sucederían las operaciones sin paréntesis:

	(1) (2) (3)	
	100 * 3600 / 9.8 * 1000	
(1)	360000 / 9.8 * 1000	1.º El primer producto.
(2)	36734.6938 * 1000	2.º El cociente.
(3)	36734693.8	3.º El segundo producto.

Al final hemos multiplicado por 1000, cuando en la fórmula del enunciado lo que hay que hacer es dividir por 1000.

Poniendo el denominador entre paréntesis se soluciona todo, pues la operación del paréntesis es la primera que se efectúa.

15. FRACCION (4.3)

Haz dos programas que escriban el valor decimal de la fracción.

$$F = \frac{\dfrac{3}{5} + \dfrac{7}{4}}{\dfrac{9}{8} + \dfrac{5}{2}}$$

En el primer programa llamas N al numerador y D al denominador, y después calculas F = N/D. En el segundo puedes hallar F directamente, sin variables intermedias (¡cuidado con los paréntesis!).

Comprueba que en ambos casos obtienes el mismo resultado.

PRIMER PROGRAMA

```
10 LET N=3/5+7/4
20 LET D=9/8+5/2
30 LET F=N/D
40 PRINT F
50 END
```

SEGUNDO PROGRAMA

```
10 LET F=(3/5+7/4)/(9/8+5/2)
20 PRINT F
30 END
```

En el segundo programa hacen falta los paréntesis para que el ordenador haga las operaciones en el orden que te interesa. Si no los pones, la máquina hace primero todas las divisiones (de izquierda a derecha) y después las sumas, siguiendo la jerarquía de las operaciones. Si escribes:

$$F = 3/5 + 7/4/9/8 + 5/2$$

la computadora calculará la siguiente fórmula, que no tiene nada que ver con el enunciado:

$$F = \frac{3}{5} + \frac{7}{4*9*8} + \frac{5}{2}$$

Cuando se ponen los paréntesis, el primero corresponde a la N del primer programa, y el segundo a la D. Entonces el valor de F corresponde a N/D como querías.

16. LA COMPRA (4.4)

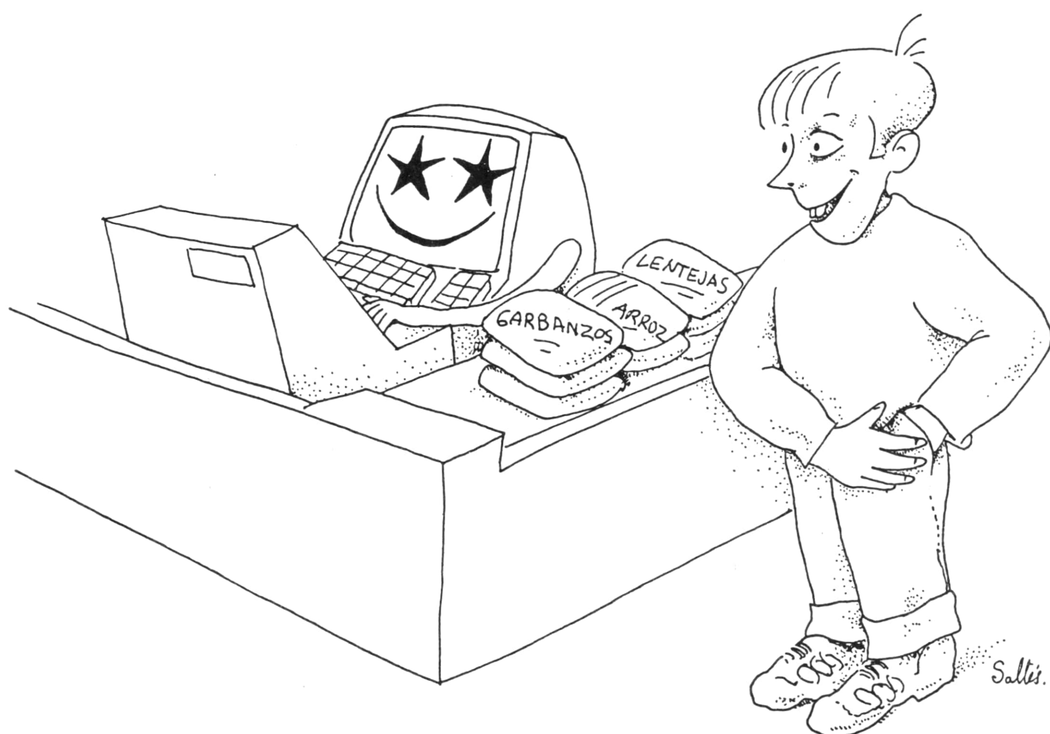
Imagina un micromercado en el que únicamente se vende arroz, garbanzos y lentejas, al precio de 117 ptas., 129 ptas. y 133 ptas. el kilo, respectivamente. La cajera tiene un microordenador, pero no tiene un programa que haga la cuenta a cada cliente. ¿Por qué no le haces tú el programa?

```
10 PRINT "TE VOY A PREGUNTAR CUANTOS KG.
   TE LLEVAS DE CADA ARTICULO"
20 PRINT "SI DE ALGUNO NO QUIERES NADA,
   CONTESTA 0"
30 PRINT
40 PRINT "CUANTOS KG. DE ARROZ TE LLEVAS"
50 INPUT A
60 LET P=117*A
70 PRINT
80 PRINT "CUANTOS KG. DE GARBANZOS"
90 INPUT G
100 LET P=P+129*G
110 PRINT
120 PRINT "Y DE LENTEJAS"
130 INPUT L
140 LET P=P+133*L
150 PRINT
160 PRINT "LA CUENTA ASCIENDE A"
170 PRINT P
180 PRINT "PESETAS"
190 END
```

En la variable P se pone al principio lo que cuesta el arroz. Si el cliente no se lleva arroz, la P vale cero. Después, en la línea 100, al precio del arroz *se le añade* el de los garbanzos:

```
LET P = P + 129*G
```

En efecto: esto es una asignación que calcula lo que hay a la derecha del signo igual (=) y lo guarda en la variable de la izquierda. Se halla el precio de los garbados y se le suma a P, que era el del arroz. El resultado es el nuevo valor de P que sustituye al anterior.



En la línea 140 se repite la misma operación con el precio de las lentejas, y la variable P contiene entonces el precio total.

El programa tiene muchas instrucciones PRINT. Hay que cuidar que estén escritas con claridad, sin dar lugar a confusión. Por eso hemos previsto en la línea 20 que, si no se desea llevar alguno de los artículos, se responda cero (0) a la pregunta correspondiente. Aunque parezca una respuesta lógica, puede haber personas que den a la tecla (RETURN) directamente, o que respondan con un mensaje como NADA, o NO QUIERO GARBANZOS. Esto produciría errores en la ejecución del programa.

Delante de cada INPUT hay una instrucción PRINT, que aclarará qué es lo que se pregunta a continuación. Además se han distribuido varias instrucciones PRINT sin nada detrás, para que impriman líneas en blanco. Todo esto es para que el programa se entienda bien y lo pueda utilizar cualquiera, aunque no sepa BASIC.

17. PITAGORAS (5.1)

Del famoso teorema de Pitágoras se deduce que "la hipotenusa es igual a la raíz cuadrada de la suma de los cuadrados de los catetos". Haz un programa que calcule la hipotenusa de triángulo rectángulo, tomando como datos los valores de los dos catetos.

Si llamamos a los catetos C1 y C2, el cálculo de la hipotenusa responde a la siguiente fórmula:

$$H = \sqrt{C1^2 + C2^2} \qquad (1)$$

Para el cálculo de la raíz cuadrada vamos a usar la función SQR (SQuare Root). El radicando ha de ir entre paréntesis, por lo que la expresión en BASIC de la fórmula (1) es:

```
LET H = SQR(C1↑2+C2↑2)
```

Mira el programa que hemos preparado:

```
10 PRINT "LONGITUD DE UN CATETO"
20 INPUT C1
30 PRINT "LONGITUD DEL OTRO CATETO"
40 INPUT C2
50 PRINT
60 LET H=SQR(C1↑2+C2↑2)
70 PRINT "LA HIPOTENUSA MIDE"
80 PRINT H
90 END
```

Como sabes, la raíz cuadrada puede escribirse como potencia de exponente fraccionario:

$\sqrt{7} = 7^{1/2}$ que en BASIC se escribe 7↑(1/2)

Esto nos abre las puertas para escribir cualquier tipo de raíces:

$\sqrt[5]{32} = 32^{1/5}$ y en BASIC pondremos 32↑(1/5)

Entonces la raíz cuadrada puede calcularse con la función SQR o con el exponente (1/2). La otra versión de la línea 60 será:

```
60 LET H=(C1↑2+C2↑2)↑(1/2)
```

18. CALIXTO EL LISTO (5.2)

Calixto es un programador listo. Ayuda a las cooperativas de productores a repartir los beneficios entre todos los socios. No cobra nada por el programa, se conforma con los céntimos de peseta que sobran en los repartos; pues, como dice, la peseta es la moneda más pequeña y menos de eso no se puede pagar.

Haz un programa para repartir un millón de pesetas de beneficios entre los 1.997 socios de una cooperativa, que diga cuánto le corresponde a cada socio y cuánto a Calixto el listo.

Para hallar el beneficio que corresponde a cada socio hay que dividir el beneficio total entre el número de socios, esto es:

$$\frac{1.000.000}{1.997} = 500{,}751127 \text{ ptas.}$$

Pero hemos quedado en que los céntimos son para Calixto. Los socios de la cooperativa se llevan la parte entera de esa cantidad, es decir:

```
B = INT(1000000/1997) = 500 pts.
```

¿Qué es lo que queda para Calixto? La diferencia entre un millón y todo lo que se llevan los socios, o sea:

```
C = 1000000 - 1997*B
```

Por abreviar la escritura de un millón, en el programa pondremos 10↑6. Sería lo mismo escribir 1E6, utilizando la notación científica.

Cuando ejecutes el programa apreciarás la astucia de Calixto, que se lleva el triple que los socios.

```
10 PRINT "BENEFICIO A REPARTIR A CADA SOC
   IO"
20 LET B=INT(10↑6/1997)
30 PRINT B
40 PRINT
50 PRINT "CANTIDAD PARA CALIXTO"
60 LET C=10↑6-1997*B
70 PRINT C
80 PRINT "QUE TIO MAS LISTO!"
90 END
```



19. DIVISION ENTERA (5.3)

Tu ordenador no sabe hallar el cociente y el resto de una división entera de dos números (de 370 dividido por 13, por ejemplo). Explícale cómo puede hacer esa división.

El ordenador hace las divisiones como las calculadoras. Si pones:

370/13

la respuesta es:

28.4615385

A veces nos interesa la división entera:

```
370 |_13_
110   28
  6
```

donde se obtiene el cociente entero y el resto. ¿Cómo calcularlos con ordenador? Vamos a utilizar la función INT. El cociente entero será:

```
INT(370/13) = INT(28.4615385) = 28
```

El resto es la diferencia de 370 menos 28*13, que da 6.

En el programa que te proponemos no pondremos esos dos números (370 y 13), sino dos variables (DN y DR) a las que podrás dar cualquier valor en los INPUT correspondientes.

```
10 PRINT "DIVIDENDO"
20 INPUT DN
30 PRINT "DIVISOR"
40 INPUT DR
50 PRINT
60 LET C=INT(DN/DR)
70 LET R=DN-C*DR
80 PRINT "COCIENTE"
90 PRINT C
100 PRINT "RESTO"
110 PRINT R
120 END
```

20. TRUNCAR NUMEROS (5.4)

El resultado de una operación puede aparecer en la pantalla con muchas cifras decimales, pero en la mayor parte de los casos nos interesa que aparezcan únicamente 2 decimales. Supongamos, como ejemplo, que el número N vale 9.12345678 y que nosotros deseamos imprimir el número 9.12 ¿Cómo quedarnos con estas cifras y prescindir de las otras cifras decimales de N? Haz estas operaciones:

1) *Multiplica N por 100:* $100 \times N = 912.345678$
2) *Aplica la función INT: INT(912.345678) = 912*
3) *Divide por 100:* $912/100 = 9.12$.

Ya tienes un número con dos decimales. Pasar del número N = 9.12345678 al número T = 9.12, se llama truncar el número N a dos cifras decimales.

Fíjate bien en las tres operaciones que has realizado y escribe una instrucción que trunque cualquier número N a dos cifras decimales.

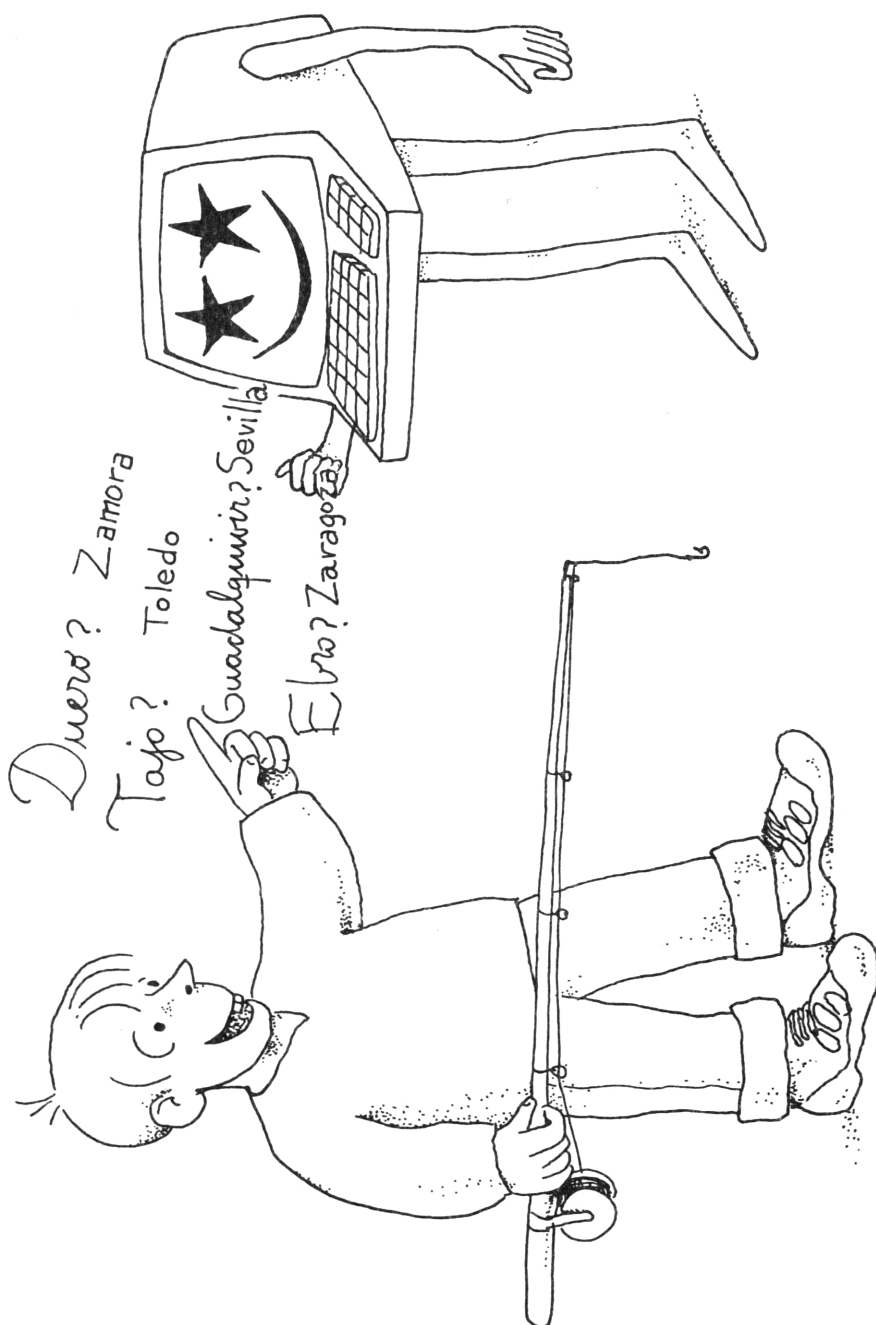
Siguiendo las orientaciones del enunciado escribimos la siguiente instrucción:

```
LET T = INT(N*100)/100
```

Queremos recalcar el orden en que se efectúan las operaciones. Como indica el enunciado, la división por 100 se efectúa después del cálculo de la parte entera (INT). Es una muestra de que todas las funciones como INT y SQR, se efectúan antes que las operaciones aritméticas (+, −, *, /,↑).

Vamos a incluir la instrucción anterior dentro de un programa que trunque cualquier número que le des a dos decimales.

```
10 PRINT "DAME UN NUMERO CON MAS DE DOS D
   ECIMALES"
20 INPUT N
30 LET T=INT(N*100)/100
40 PRINT "EL NUMERO TRUNCADO ES"
50 PRINT T
60 END
```


Duero? Zamora
Tajo? Toledo
Guadalquivir? Sevilla
Ebro? Zaragoza

21. **RIOS** (6.1)

En un programa tenemos las siguientes líneas:

```
100 DATA DUERO,TAJO,GUADALQUIVIR
110 DATA EBRO,SEVILLA,ZAMORA
120 DATA ZARAGOZA,TOLEDO
```

Haz un programa que nos escriba en líneas sucesivas el río y la ciudad por la que pasa.

Para leer los ríos y las ciudades utilizaremos las instrucciones READ, que colocaremos al principio del programa. A continuación, con las instrucciones PRINT, conseguiremos escribir los ríos y las ciudades por las que pasan. Y al final pondremos las líneas DATA.

Para los ríos utilizamos las variables A$, B$, C$ y D$ y para las ciudades W$, X$, Y$ y Z$.

```
10 READ A$,B$,C$,D$
20 READ W$,X$,Y$,Z$
30 PRINT A$,X$
40 PRINT B$,Z$
50 PRINT C$,W$
60 PRINT D$,Y$
100 DATA "DUERO","TAJO","GUADALQUIVIR"
110 DATA "EBRO","SEVILLA","ZAMORA"
120 DATA "ZARAGOZA","TOLEDO"
130 END
```

En algunos ordenadores se pueden escribir los datos sin comillas. Prueba si el tuyo lo admite.

22. NOTAS (6.2)

Prepara un programa que lea (READ) los nombres de tres compañeros (DATA); que también lea sus notas de matemáticas; y que, por último, escriba los nombres junto con la nota correspondiente.



Los nombres de tus compañeros (Francisco Pérez, Joaquín López y Manuel Fernández) los ponemos en la instrucción DATA, que colocamos al principio del programa. Sus notas de matemáticas 7, 9 y 5 también irán en una línea DATA. En otro bloque colocamos las instrucciones READ con las variables A$, B$ y C$ para los nombres y A, B y C para las notas. Y como tercer bloque escribimos las instrucciones PRINT, de modo que cada compañero aparezca con la nota correspondiente.

```
10 DATA "FRANCISCO PEREZ","JOAQUIN LOPEZ",
   "MANUEL FERNANDEZ"
20 DATA 7,9,5
30 READ A$,B$,C$
40 READ A,B,C
50 PRINT A$,A
60 PRINT B$,B
70 PRINT C$,C
80 END
```

23. FECHAS DE NACIMIENTO (6.3)

Guarda en líneas DATA los nombres y fechas de nacimiento de los miembros de tu familia. Completa el programa para que lea los datos y los escriba ordenadamente.

Tu padre se llama JAVIER (20-2-40), tu madre DOLORES (3-7-45) y tienes dos hermanos que se llaman JUAN (15-12-67) y ANA (30-4-70). Y tu nombre es JOSE (23-1-69).

Los nombres y las fechas de nacimiento serán almacenados en las instrucciones DATA para que sean leídos por las correspondientes instrucciones READ.

En las variables A$, B$, C$, D$ y E$ se guardan los nombres y en V$, W$, X$, Y$ y Z$ las fechas de nacimiento.

Delante de cada fecha ponemos un espacio en blanco, para que al imprimir no se junte el nombre con la fecha.

```
10 READ A$,V$
20 READ B$,W$
30 READ C$,X$
40 READ D$,Y$
50 READ E$,Z$
60 DATA "JAVIER"," 20-2-40"
70 DATA "DOLORES"," 3-7-45"
80 DATA "JUAN"," 15-12-67"
90 DATA "ANA"," 30-4-70"
100 DATA "JOSE"," 23-1-69"
110 PRINT A$;V$
120 PRINT B$;W$
130 PRINT C$;X$
140 PRINT D$;Y$
150 PRINT E$;Z$
160 END
```

24. DIRECCION EN UN SOBRE (6.4)

En dos líneas DATA pon el nombre y la dirección de dos amigos. Diseña un programa para que lea los datos y los escriba en forma de dirección postal (como si la pantalla fuera un sobre).

Uno de los amigos se llama JUAN PEREZ GOMEZ, y su dirección es CERCEDILLA, 120. 28015-Madrid. El otro, JOSE LOPEZ FERNANDEZ, vive en OCA, 625. 28025-MADRID. Para el nombre, la calle, y el código postal y la ciudad, utilizamos las variables N$, D$ y P$ respectivamente.

```
10 READ N$,D$,P$
20 PRINT "SR. D. ";N$
30 PRINT
40 PRINT
50 PRINT D$
60 PRINT
70 PRINT
80 PRINT P$
90 PRINT
100 PRINT
110 READ N$,D$,P$
120 PRINT "SR. D. ";N$
130 PRINT
140 PRINT
150 PRINT D$
160 PRINT
170 PRINT
180 PRINT P$
190 DATA "JUAN PEREZ GOMEZ","CERCEDILLA,
    120","28015-MADRID"
200 DATA "JOSE LOPEZ FERNANDEZ","OCA, 625
    ","28025-MADRID"
210 END
```

Si ejecutas el programa, sale en la pantalla:

```
SR. D. JUAN PEREZ GOMEZ

CERCEDILLA, 120

28015-MADRID

SR. D. JOSE LOPEZ FERNANDEZ

OCA, 625

28025-MADRID
```





25. EL ORDENADOR EDUCADO (7.1)

Debes acostumbrar a tu ordenador a que sea educado con tus amigos. Prográmalo para que pregunte:

```
COMO TE LLAMAS?
```

y que, inmediatamente de recibir el nombre de tu amigo (LUIS), conteste:

```
ME ALEGRO DE CONOCERTE LUIS
```

El programa es así de fácil:

```
10 REM EL ORDENADOR EDUCADO
20 PRINT "COMO TE LLAMAS";
30 INPUT N$
40 PRINT "ME ALEGRO DE CONOCERTE ";N$
50 END
```

Con la línea 20 el ordenador escribe COMO TE LLAMAS.

La 30 sirve para introducir en N$ el nombre de tu amigo (LUIS).

La 40 hace que el ordenador escriba en la pantalla:

```
ME ALEGRO DE CONOCERTE LUIS
```

El punto y coma final de la línea 20 está puesto para que la interrogación del INPUT salga pegada a COMO TE LLAMAS.

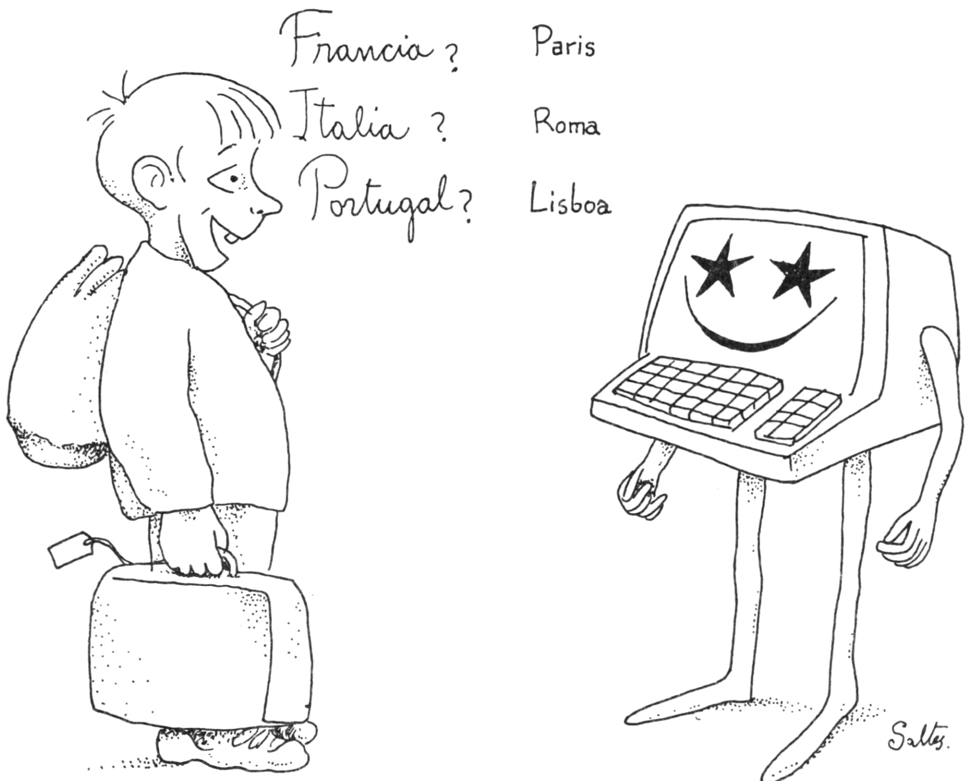
26. CAPITALES (7.2)

Utiliza instrucciones PRINT con coma (,) para que tu ordenador escriba:

```
FRANCIA    PARIS
ITALIA     ROMA
PORTUGAL   LISBOA
```

La coma (,) en una instrucción PRINT sirve para separar las salidas dentro de una misma línea, como puedes observar al ejecutar el programa.

```
10 REM CAPITALES
20 PRINT "FRANCIA","PARIS"
30 PRINT "ITALIA","ROMA"
40 PRINT "PORTUGAL","LISBOA"
50 END
```



27. DIBUJO UN PEZ (7.3)

Haz uso de la función TAB para que el pez del ejercicio 1 quede centrado en la pantalla.

```
10 REM DIBUJO UN PEZ
20 REM BAJAR EL CURSOR NUEVE LINEAS
30 PRINT:PRINT:PRINT
40 PRINT:PRINT:PRINT
50 PRINT:PRINT:PRINT
100 REM CENTRAR EL PEZ
110 PRINT TAB(10);"    * * * *"
120 PRINT TAB(10);"  * *       *   *"
130 PRINT TAB(10);"*   *         * *"
140 PRINT TAB(10);"  * *       *   *"
150 PRINT TAB(10);"    * * * *"
160 END
```

Hemos hecho un programa que dibuja el pez en el centro de la pantalla.

Las líneas 30, 40 y 50 dejan nueve líneas en blanco antes de comenzar a dibujar el pez.

La instrucción PRINT TAB(10) hace que el ordenador comience a imprimir en la posición 10 de la línea correspondiente.

Como habrás observado, se pueden poner varias instrucciones en una misma línea de programa, siempre que se separen por dos puntos (:). Esto es lo que hemos hecho en las líneas 30, 40 y 50.

Para el Spectrum puedes utilizar este programa:

```
10 REM DIBUJO UN PEZ
20 PRINT AT 9,10;"    * * * *"
30 PRINT AT 10,10;"  * *       *   *"
40 PRINT AT 11,10;"*   *         * *"
50 PRINT AT 12,10;"  * *       *   *"
60 PRINT AT 13,10;"    * * * *"
70 STOP
```

28. **SOBRE** (7.4)

Prepara un programa que pida el nombre y la dirección postal de una persona (calle, número, código postal y población). Y que luego escriba (TAB) en forma de sobre:

```
MANUEL DIAZ RUIZ
TRAFALGAR 123
28010 MADRID
      ESPAÑA
```

```
   10 REM SOBRE
(1)20 (BORRAR LA PANTALLA)
   30 PRINT "NOMBRE";:INPUT N$
   40 PRINT "CALLE Y NUMERO";:INPUT C$
   50 PRINT "CODIGO POSTAL Y POBLACION";:INP
      UT P$
(1)60 (BORRAR LA PANTALLA)
   70 REM BAJAR NUEVE LINEAS
(2)80 PRINT:PRINT:PRINT
   90 PRINT:PRINT:PRINT
   100 PRINT:PRINT:PRINT
   110 REM CENTRAR EL SOBRE
   120 PRINT TAB(10);N$
   130 PRINT TAB(10);C$
   140 PRINT TAB(10);P$
   150 PRINT TAB(16);"ESPANA"
   160 END
```

El programa consta de dos bloques. El primero (líneas 20-50) permite introducir los datos en las variables N$, C$ y P$. El segundo bloque (líneas 60-150) borra la pantalla, baja el cursor nueve líneas y escribe el sobre a partir de la posición 10.

---

(1) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT " [SHIFT] [CLR HOME] ".

(2) Si tu ordenador es un Spectrum, sustituye las cuatro líneas por ésta: 70 PRINT AT 9,0.

29. ENCUENTRA LOS ERRORES (8.1)

En un programa aparece esta línea:

```
20 INPUT "NOMBRE, CIUDAD, CALLE";N$;CI;CA
```

¿Observas en ella algún error?

Un primer error que se observa es que las variables CI y CA, que van a guardar el nombre de la ciudad y de la calle respectivamente, no llevan el símbolo **$**.

Otro error es que las tres variables deben ir separadas por comas, y no por puntos y comas. La línea correcta es:

```
20 INPUT "NOMBRE, CIUDAD, CALLE";N$,CI$,C
   A$
```

En el Spectrum, sin embargo, los errores serían de otro tipo: En primer lugar, los nombres de las variables de cadena sólo pueden constar de una letra y el símbolo **$**, por supuesto. Por otro lado, las variables pueden separarse por comas y también por puntos y comas.

Estas dos líneas son correctas en el Spectrum:

```
20 INPUT "NOMBRE, CIUDAD, CALLE";N$,C$,K$
```

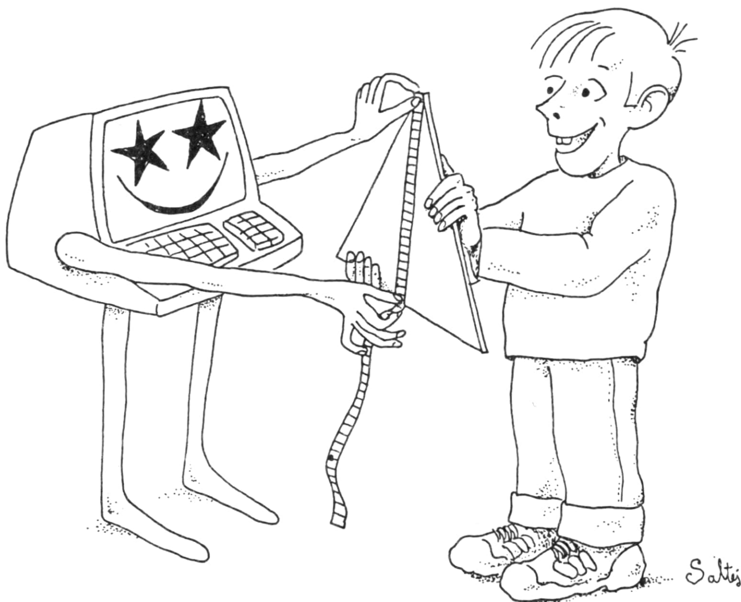
```
20 INPUT "NOMBRE, CIUDAD, CALLE";N$;C$;K$
```

30. AREA DE UN TRIANGULO (8.2)

Prepara un programa para que el ordenador calcule el área de un triángulo. Para meter los datos utiliza los INPUT con etiqueta.

El programa se compone de tres partes. La primera nos pide los datos. La segunda hace los cálculos mediante la fórmula del área del triángulo, que como debes recordar es b·a/2. Y la tercera presenta el resultado. Para que se entienda mejor ponemos instrucciones REM en el programa.

```
10 REM AREA DE UN TRIANGULO
20 REM PETICION DE DATOS
30 INPUT "DAME LA BASE";B
40 PRINT
50 INPUT "DAME LA ALTURA";A
60 PRINT
70 REM CALCULO DEL AREA
80 LET S=B*A/2
90 REM PRESENTACION DEL RESULTADO
100 PRINT "EL AREA ES ";S
110 END
```



31. AGENDA (8.3)

Haz un programa para que el ordenador te pida (INPUT) el NOMBRE, APELLIDOS, NUMERO DE TELEFONO y ESTADO CIVIL de una persona. Debes colocar etiquetas en las líneas INPUT para que sepas qué te está pidiendo el ordenador. Completa el programa para que escriba esos datos en la pantalla.

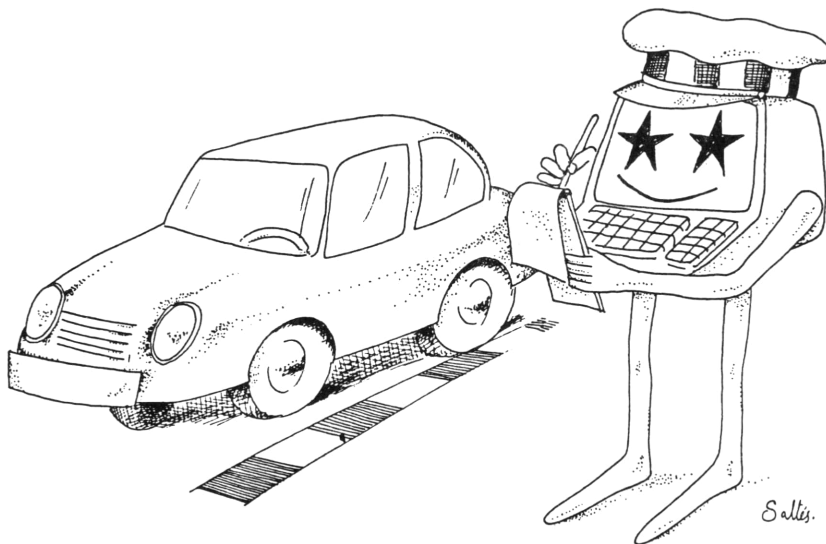
El programa consta de dos bloques. Uno es para entrada de datos y otro, para salida de datos. Dos instrucciones REM dirán dónde comienza cada uno de ellos.

En este programa volvemos a utilizar los dos puntos, que sirven como recordarás, para escribir varias instrucciones en una misma línea.

```
10 REM DATOS DE UNA PERSONA
20 REM ENTRADA DE DATOS
30 INPUT "DAME EL NOMBRE";N$
40 PRINT
50 INPUT "DAME LOS APELLIDOS";A$
60 PRINT
70 INPUT "DAME EL NUMERO DE TELEFONO";T$
80 PRINT
90 INPUT "DAME EL ESTADO CIVIL";E$
100 PRINT
110 REM SALIDA DE DATOS
120 PRINT A$,N$:PRINT
130 PRINT "TFNO: ";T$:PRINT
140 PRINT E$
150 END
```

32. FICHA DE UN COCHE (8.4)

Diseña un programa para que el ordenador te pida la MARCA, el MODELO y la MATRICULA de un coche (todo en una sola línea INPUT) y a continuación que lo escriba en tres líneas.



Para realizar el programa de este ejercicio recuerda que en una instrucción INPUT puedes poner varias variables, siempre que las separes por comas (ver ejercicio 29). El programa es el siguiente:

```
   10 REM FICHA DE UN COCHE
   20 PRINT "DATOS DE UN COCHE":PRINT
(1)30 INPUT "MARCA, MODELO, MATRICULA";M1$,M
      2$,M3$
   40 PRINT
(1)50 PRINT "MARCA: ";M1$
   60 PRINT
(1)70 PRINT "MODELO: ";M2$
   80 PRINT
(1)90 PRINT "MATRICULA: ";M3$
   100 END
```

---

(1) En el Spectrum las variables de cadena sólo pueden tener una letra. Para este programa puedes utilizar, por ejemplo, las variables M$, N$ y P$.

33. VOY AL COLE (9.1)

Escribe en forma de algoritmo lo que haces desde que te levantas hasta que vas al colegio.

Esto es lo que hace un niño "modelo" cuando se levanta.

1. Me desperezo según voy al cuarto de baño.
2. Me ducho.
   - 2.1. Me quito el pijama y me meto en la ducha.
   - 2.2. Me ducho.
   - 2.3. Me seco.
   - 2.4. Me peino.
3. Me visto.
   - 3.1. Cojo la ropa y me la pongo.
   - 3.2. Me calzo.
4. Desayuno.
   - 4.1. Preparo las tostadas.
   - 4.2. Me las como.
   - 4.3. Cojo la taza y me sirvo la leche.
   - 4.4. Le añado cacao y tras batirlo durante algún tiempo me lo tomo.
5. Me voy al colegio.

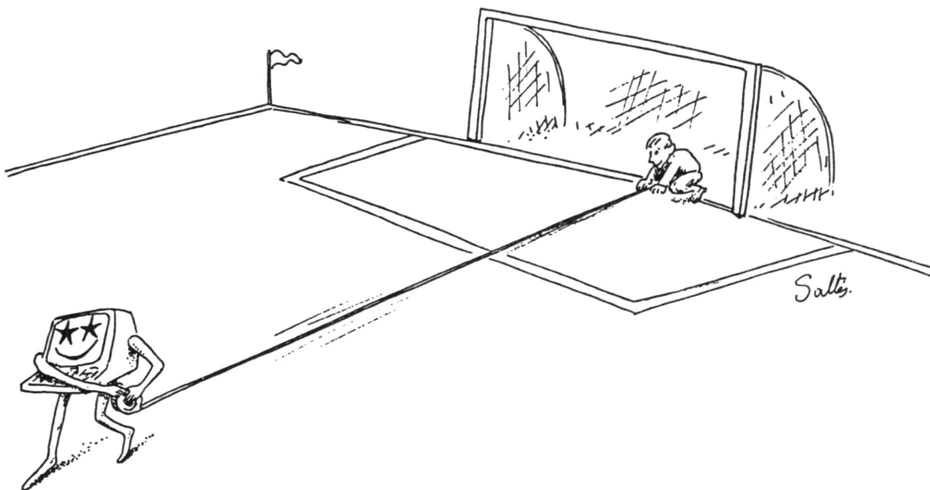


34. CAMPO DE FUTBOL (9.2)

Dado un campo de fútbol escribe en forma de algoritmo las acciones que hay que realizar para obtener su área.

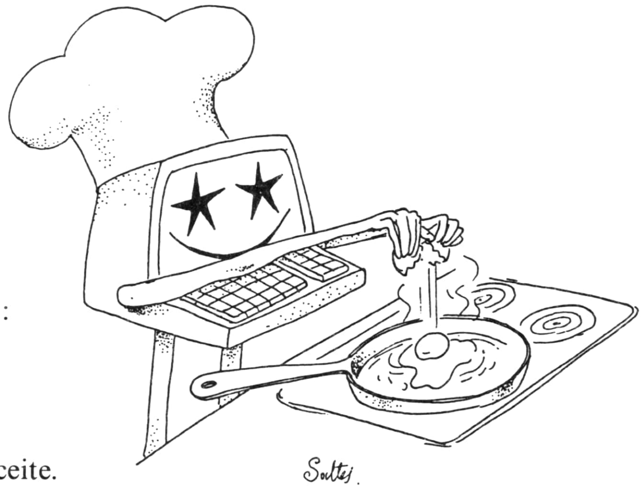
Un algoritmo puede ser el siguiente:

1. Tomamos una cinta métrica.
2. Medimos el largo del campo.
3. Escribimos la medida.
4. Medimos el ancho del campo.
5. Escribimos la medida.
6. Multiplicamos la medida del largo por la medida del ancho.
7. Escribimos el resultado obtenido, que es el área del campo de fútbol.



35. FREIR UN HUEVO (9.3)

Freir un huevo requiere una serie de operaciones. Escríbelas en forma de algoritmo.



Una receta puede ser ésta:

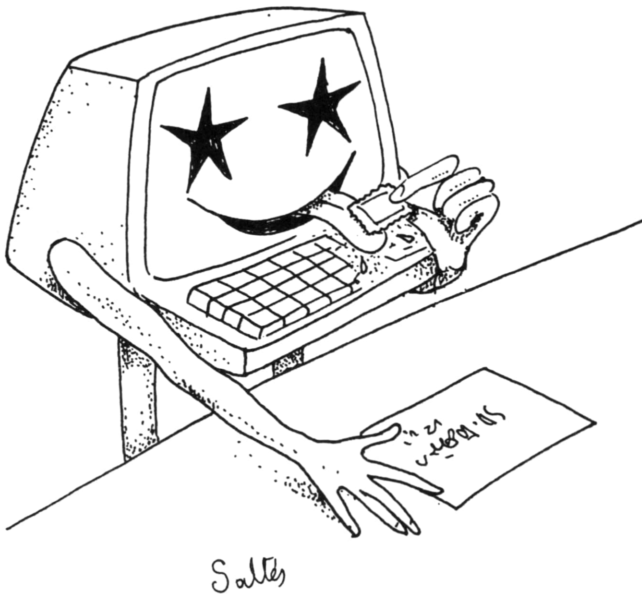
1. Preparo la sartén.
2. Cojo el aceite.
3. Lleno la sartén de aceite.
4. Enciendo el fuego de la cocina.
5. Pongo la sartén al fuego.
6. Saco el huevo de la nevera.
7. Veo si sale humo de la sartén, lo que indica que el aceite está caliente.
8. Casco el huevo y lo echo a la sartén.
9. Cojo la paleta.
10. Introduzco la paleta en el aceite y echo un poco de aceite sobre el huevo.
11. Cojo el plato.
12. Saco el huevo con la paleta y lo pongo en el plato.

36. PEGAR UN SELLO (9.4)

Escribe un algoritmo para pegar un sello de correos en un sobre.

El algoritmo puede ser:

1. Cojo el sobre.
2. Cojo el sello.
3. Humedezco la cara engomada.
4. Lo llevo a la esquina superior derecha del sobre.
5. Lo apoyo sobre el sobre.
6. Aprieto con la mano.



37. ASTERISCOS (10.1)

Mira a ver qué efectos producen estos programas.

```
10 PRINT "*"
20 GOTO 10
```

```
10 PRINT "*";
20 GOTO 10
```

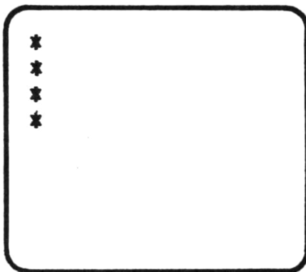
```
10 PRINT " *"
20 GOTO 10
```

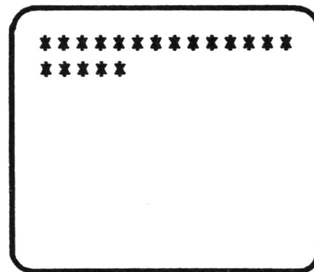
```
10 PRINT " *";
20 GOTO 10
```

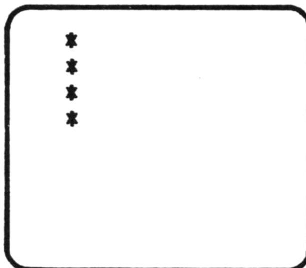
En los cuatro programas tienes una línea 20 que dice GOTO 10. Ya sabes que esta instrucción es un salto a la línea número 10. Por esta razón el programa no acaba, y tienes que pulsar la tecla (STOP), si quieres que se pare.

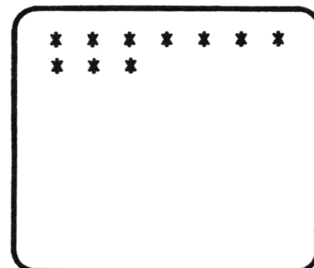
Además, dos de los programas acaban las líneas PRINT con punto y coma, para que los asteriscos salgan pegados.

El espacio en blanco que se deja en dos de ellos también afecta a lo que aparece en la pantalla:



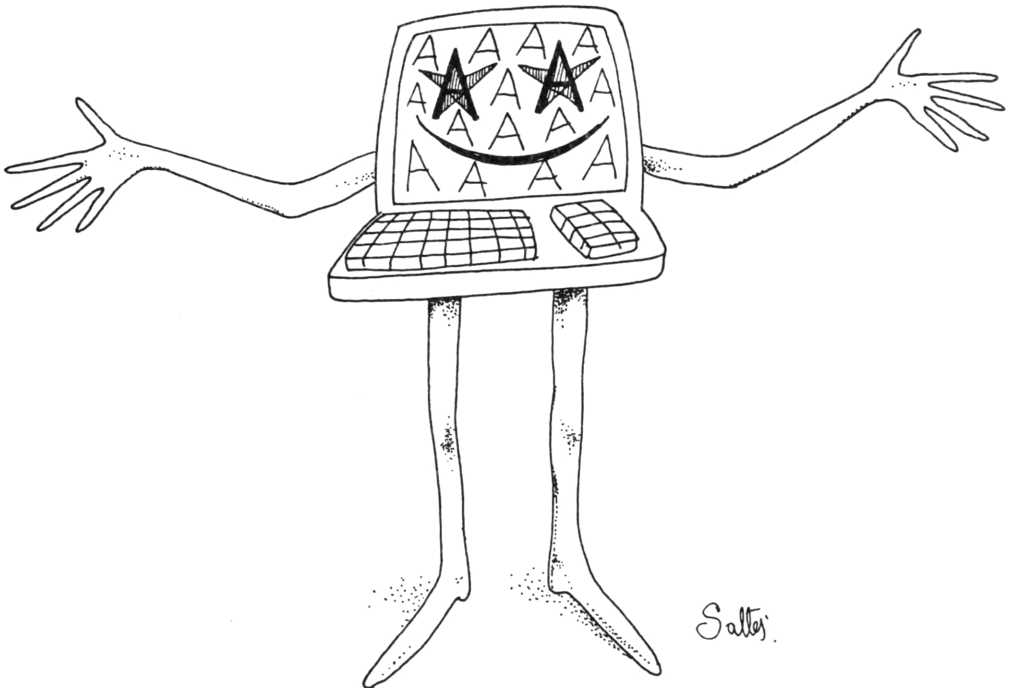






38. AAA... (10.2)

Llena toda la pantalla de letras A.



El programa de este ejercicio es muy parecido a uno de los que están en el ejercicio 37.

No tienes más que cambiar la línea PRINT "*"; por la PRINT "A"; .

El programa es:

```
10 REM AAA...
20 PRINT "A";
30 GOTO 20
40 END
```

Aunque en la línea 40 hemos puesto END, ya sabes que este programa no acaba jamás, por lo que debes pulsar la tecla (STOP) .

39. BUEN PROGRAMADOR (10.3)

Haz un programa que escriba muchas veces:

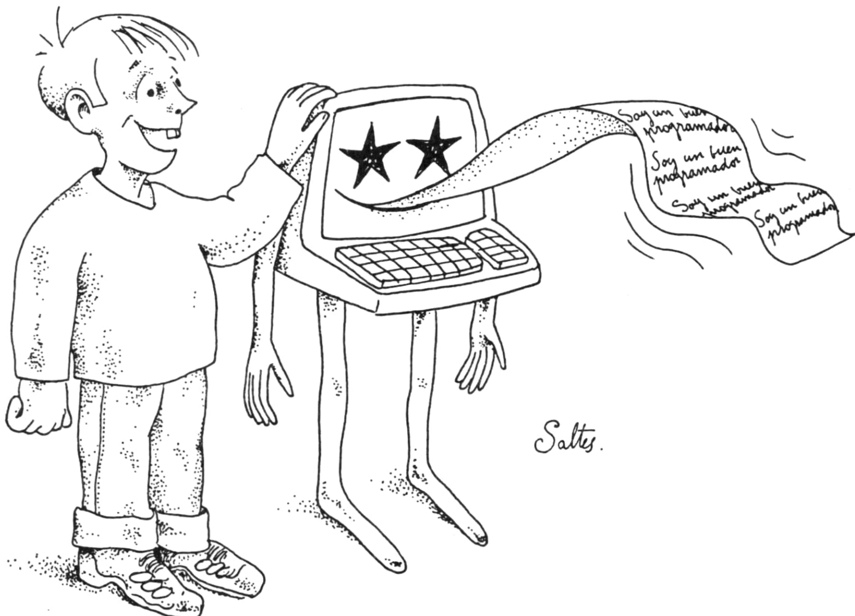
SOY BUEN PROGRAMADOR

Para hacer el programa de este ejercicio fíjate en uno de los programas del ejercicio 37, ya que se trata de que salga muchas veces SOY BUEN PROGRAMADOR, lo que te indica que debes hacer uso de la instrucción GOTO, para volver a la línea donde has escrito la frase deseada.

Así resulta el programa:

```
10 REM BUEN PROGRAMADOR
20 PRINT "SOY BUEN PROGRAMADOR"
30 GOTO 20
40 END
```

La línea 40 lleva la instrucción END, pero ya sabes que el programa no se detiene, por lo que debes pulsar la tecla (STOP).



40. TODOS LOS NUMEROS PARES (10.4)

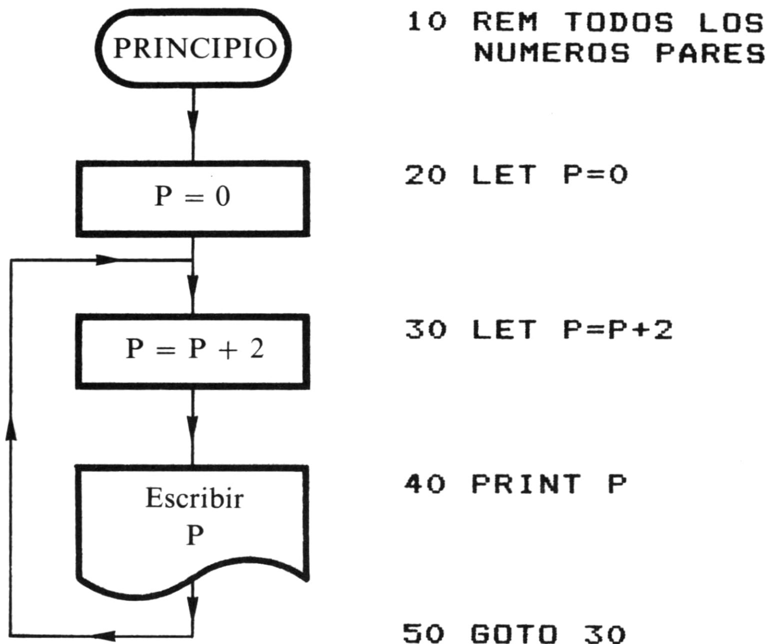
Haz un programa para que el ordenador escriba números pares.

La idea clave es establecer una variable P que vaya conteniendo sucesivamente los números pares 2, 4, 6, 8, ...

Esto se consigue con un contador:

```
P = P + 2
```

que aumenta el valor de P en dos unidades cada vez que se ejecuta esa instrucción. Previamente hay que poner el contador a cero, a lo que se llama *inicializar* la variable P.

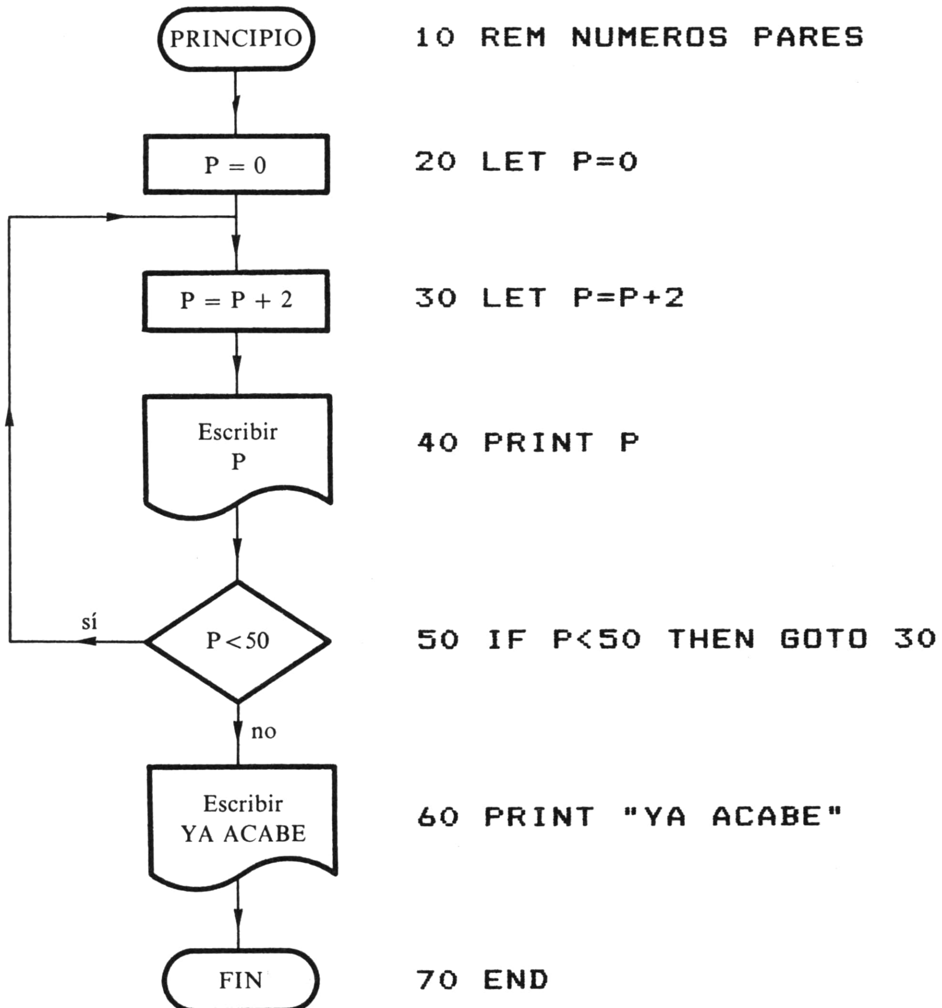


41. NUMEROS PARES (11.1)

Prepara un programa que escriba los números pares, de modo que al llegar al 50 se detenga y escriba YA ACABE.

La primera parte de este programa es igual que la del ejercicio 40.

Ahora, como queremos que el último número que escriba sea el 50, tenemos que controlar el valor de P; de modo que, si P<50, el ordenador salte a la línea 30, y si no se cumple esa condición (P ya es 50), escriba YA ACABE y termine.




```
10 REM NUMEROS PARES
20 LET P=0
30 LET P=P+2
40 PRINT P
50 IF P<50 THEN GOTO 30
60 PRINT "YA ACABE"
70 END
```

42. AUTORIZACION (11.2)

Para impedir que personas no autorizadas utilicen tus programas, puedes incluir al principio de ellos unas líneas que pidan al usuario una clave secreta. Si éste da la clave correcta, el programa continúa. Si no, saca el mensaje: LO SIENTO. NO ESTA AUTORIZADO, y el programa termina.

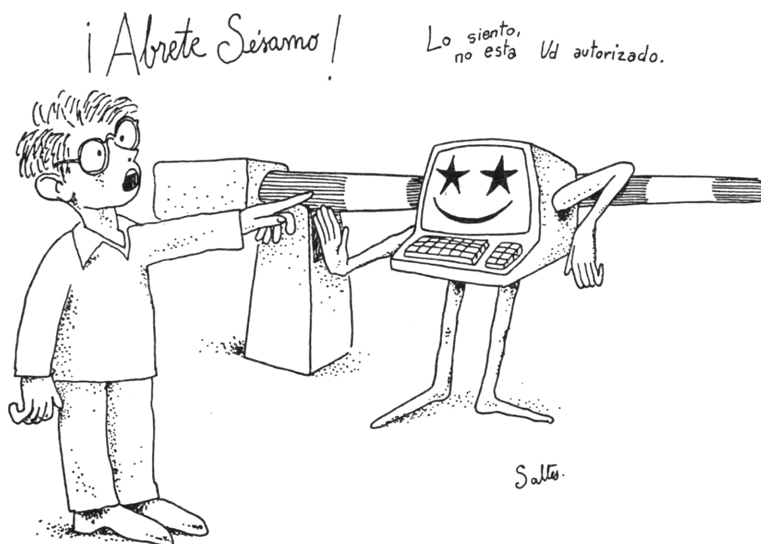
Programa estas primeras líneas con la palabra clave ABRACADABRA.

```
10 REM AUTORIZACION
20 INPUT "DIME LA CLAVE SECRETA";C$
30 IF C$<>"ABRACADABRA" THEN PRINT "LO SI
   ENTO. NO ESTA AUTORIZADO":END
40 PRINT "VALE"
50 REM AQUI CONTINUA EL PROGRAMA
```

La línea 20 tiene una doble misión: por un lado, imprimir en la pantalla DIME LA CLAVE SECRETA, y por otro, esperar a que demos la clave y guardarla en C$.

En la línea 30 el ordenador comprueba si C$ *no es* ABRACADABRA, en cuyo caso imprime LO SIENTO. NO ESTA AUTORIZADO, y termina; si no (C$ es ABRACADABRA), pasa a la línea siguiente y escribe VALE.

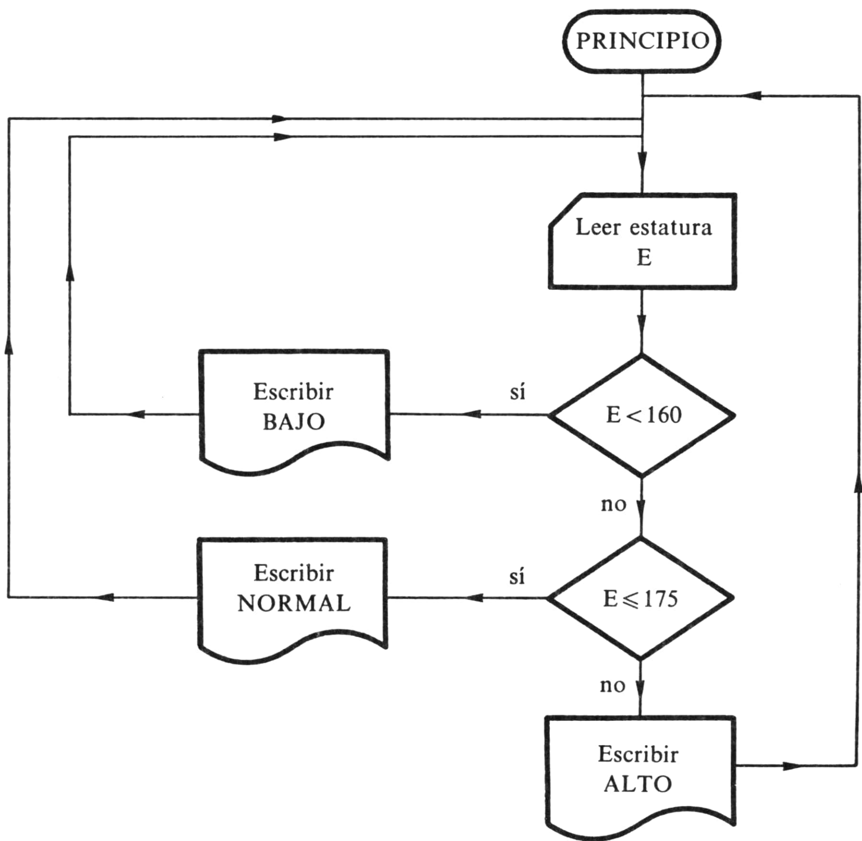
Como ves, la condición de IF-THEN puede referirse también a palabras.



43. ESTATURAS (11.3)

Una empresa clasifica a su personal según la estatura. Un empleado es BAJO cuando mide menos de 160 cm., NORMAL cuando su estatura va desde 160 hasta 175 cm., y ALTO cuando mide más de 175 cm. La empresa necesita preparar a su ordenador para que acepte como dato la estatura y escriba BAJO, NORMAL o ALTO, según corresponda. ¿Quieres hacer el programa?

El ordenador va a tomar la estatura de cada persona mediante una instrucción INPUT y la va a guardar en la variable E. Si el contenido de E es menor que 160, escribirá BAJO y volverá a pedir la estatura de la persona siguiente. Si no, probará a ver si E es menor o igual que 175, en cuyo caso, escribirá NORMAL y volverá a pedir otra estatura. Si no, escribirá ALTO y pedirá la estatura de la siguiente persona. Este puede ser el diagrama de flujo:



```
10 REM ESTATURAS
20 PRINT
30 INPUT "ESTATURA EN CENTIMETROS ";E
40 IF E<160 THEN PRINT "BAJO":GOTO 20
50 IF E<=175 THEN PRINT "NORMAL":GOTO 20
60 PRINT "ALTO":GOTO 20
```

En la línea 20 hemos puesto una instrucción PRINT (sin nada) para que queden separados los datos de cada persona de los de la anterior.

44. FACTORES PRIMOS (11.4)

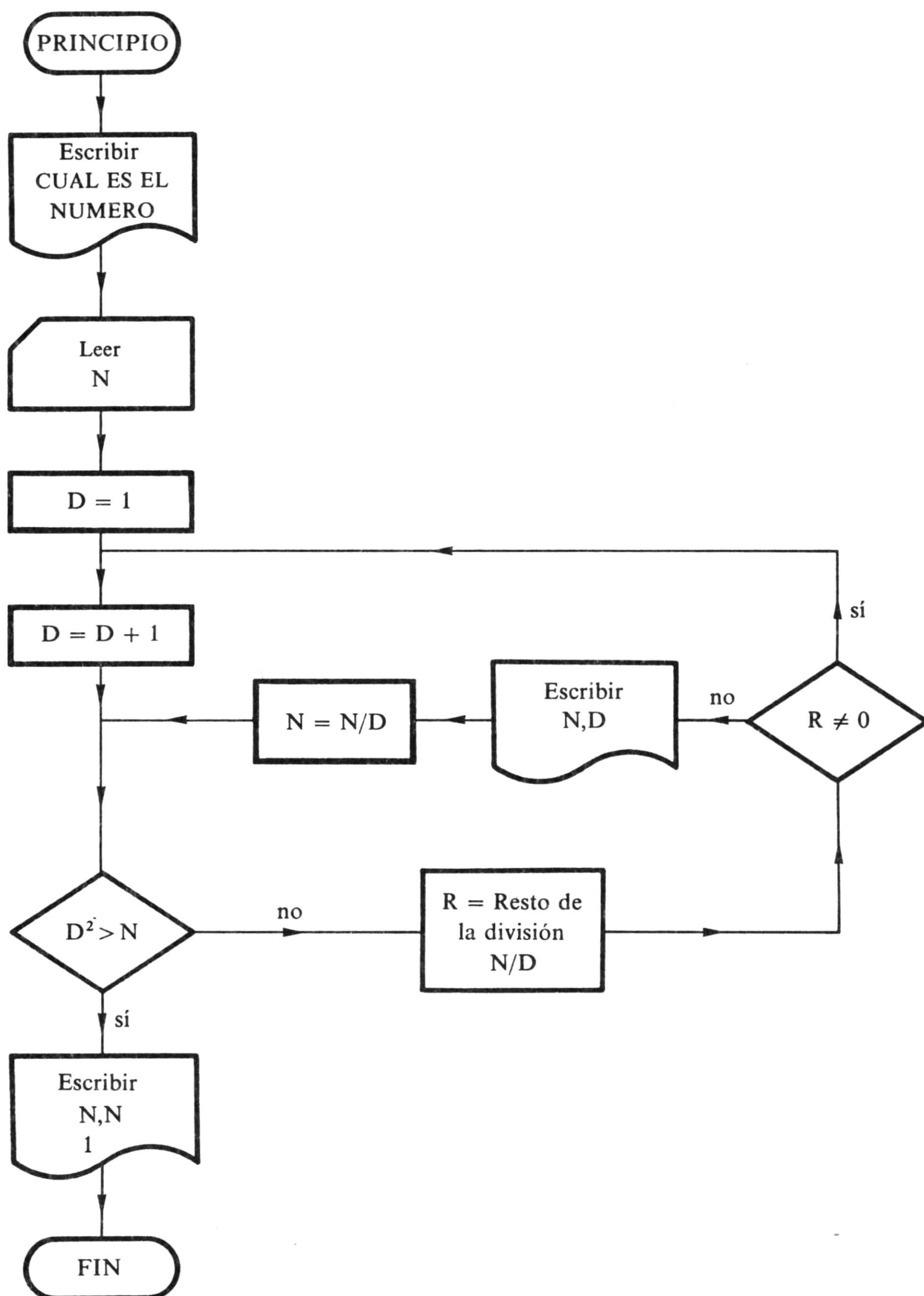
Este programa sirve para descomponer un número N en sus factores primos. Pruébalo en tu ordenador. Haz el diagrama de flujo que le corresponde.

```
10 REM FACTORES PRIMOS
20 PRINT "CUAL ES EL NUMERO";
30 INPUT N
40 LET D=1
50 LET D=D+1
60 IF D*D>N THEN GOTO 120
70 LET R=N-D*INT(N/D)
80 IF R<>0 THEN GOTO 50
90 PRINT N,D
100 LET N=N/D
110 GOTO 60
120 PRINT N,N
130 PRINT 1
140 END
```

Mira lo que queda escrito en la pantalla cuando, por ejemplo, descompones el número 72 en sus factores primos:

```
CUAL ES EL NUMERO? 72
 72            2
 36            2
 18            2
 9             3
 3             3
 1
```

Sólo falta la rayita vertical para que el ordenador lo haga como tú. ¡Intenta añadirla! Para ello modifica las líneas 90 y 120.



PRINCIPIO
Escribir
CUAL ES EL
NUMERO
Leer
N
D = 1
D = D + 1
sí
Escribir
N,D
no
R ≠ 0
N = N/D
no
R = Resto de
la división
N/D
D² > N
sí
Escribir
N,N
1
FIN

45. IMITACION DE UN DADO (12.1)

Haz un programa que imite una tirada de un dado.

La instrucción clave para imitar el dado es:

```
LET D = INT(6*RND(1))+1
```

que guarda en la variable D un número tomado al azar de entre éstos: 1, 2, 3, 4, 5 y 6.

Expliquemos cómo escoge el ordenador ese número:

Primero halla RND(1), que es un número decimal comprendido entre 0 y 1, sin ser 0 ni 1, obtenido "a suertes".

Después multiplica por 6, y obtiene un número decimal comprendido entre 0 y 6, sin ser 0 ni 6.

A continuación halla la parte entera, y tiene uno de estos números: 0, 1, 2, 3, 4 y 5.

Por último suma 1, para obtener como resultado uno de los números: 1, 2, 3, 4, 5 y 6, que guarda en D.

```
   10 REM UNA TIRADA DE UN DADO
(1)20 LET D=INT(6*RND(1))+1
   30 PRINT D
   40 END
```

(1) Si tu ordenador es un Spectrum, sustituye la línea 20 por éstas dos:

```
15 RANDOMIZE
20 LET D=INT(6*RND)+1
```

Este otro programa ofrece la posibilidad de obtener muchas tiradas de un dado:

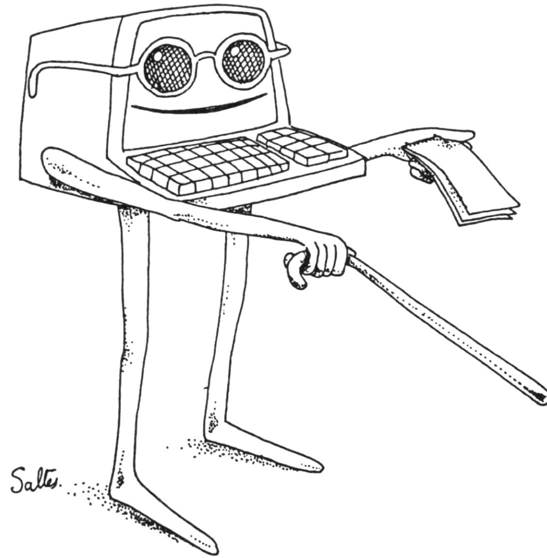
```
   10 REM MUCHAS TIRADAS DE UN DADO
   20 PRINT
   30 INPUT "LANZAS EL DADO (SI/NO)";R$
(1)40 IF R$="NO" THEN END
(2)50 LET D=INT(6*RND(1))+1
   60 PRINT TAB(10);D
   70 GOTO 20
```

(1) Para Sinclair cambiar END por STOP.

(2) Para Sinclair tienes que efectuar los cambios análogos a los de la línea 20 del programa anterior.

46. EL SORTEO DE LA ONCE (12.2)

Diseña un programa que imite el sorteo del cupón pro-ciegos. Tienes que obtener a suertes un número del 0 al 9999.



Como el valor de RND(1) es un número decimal elegido a suertes comprendido entre 0 y 1 (sin ser 0 ni 1), el número 10000*RND(1) estará comprendido entre 0 y 10000 (sin ser 0 ni 10000).

La parte entera de 10000*RND(1), que en BASIC se escribe INT(10000*RND(1)), será alguno de los número enteros 0, 1, 2, 3, ..., 9999. Esta última expresión es la que nos va a proporcionar el número premiado.

```
   10 REM EL SORTEO DE LA ONCE
(1)20 LET N=INT(10000*RND(1))
   30 PRINT "EL NUMERO PREMIADO EN EL SORTEO
      DE LA ONCE HA SIDO EL ";N
   40 END
```

(1) Si tu ordenador es un Spectrum, sustituye la línea 20 por estas dos:

```
20 RANDOMIZE
25 LET N=INT(10000*RND)
```

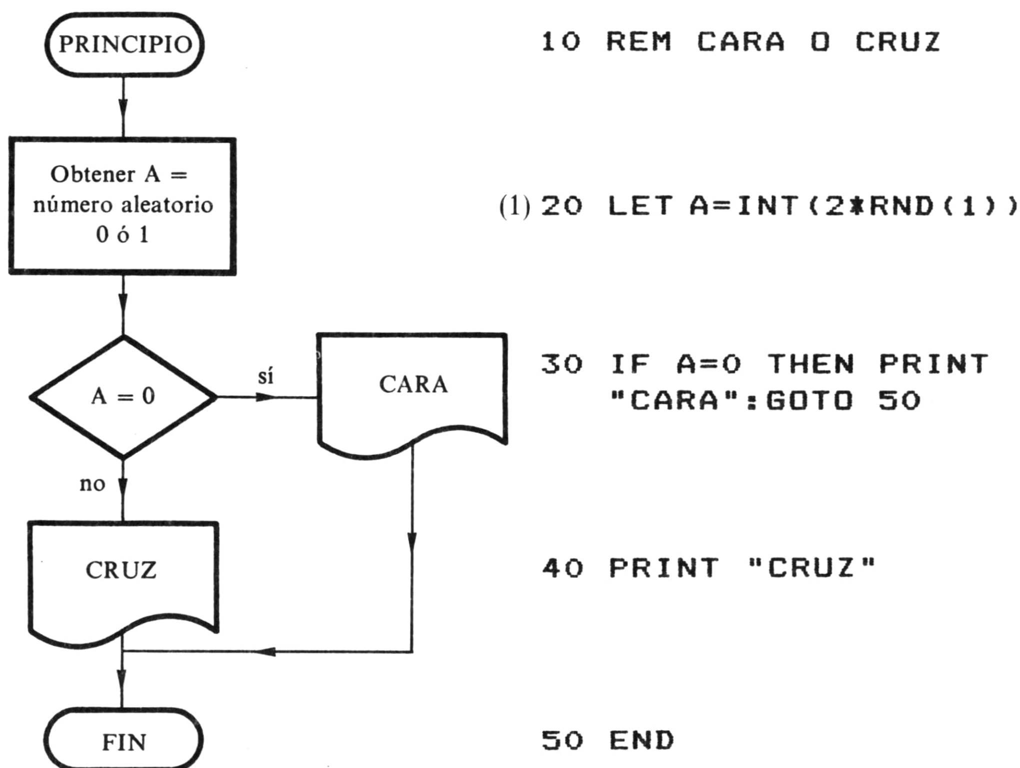
47. CARA O CRUZ (12.3)

Haz un programa que imite el lanzamiento de una moneda. Para ello tendrás que generar los números aleatorios 0 y 1 con la instrucción:

```
LET A = INT(2*RND(1))
```

Luego tendrás que hacer corresponder al 0 la cara y al 1 la cruz.

Después de ejecutarse la instrucción LET A=INT(2*RND(1)), en la variable A estará el número 0 ó el número 1. Si está el cero, haremos que el ordenador escriba la palabra CARA; si no está el cero, haremos que escriba la palabra CRUZ.




```
10 REM CARA O CRUZ
(1) 20 LET A=INT(2*RND(1))
30 IF A=0 THEN PRINT
   "CARA":GOTO 50
40 PRINT "CRUZ"
50 END
```

(1) Si tu ordenador es un Spectrum, sustituye la línea 20 por estas dos:

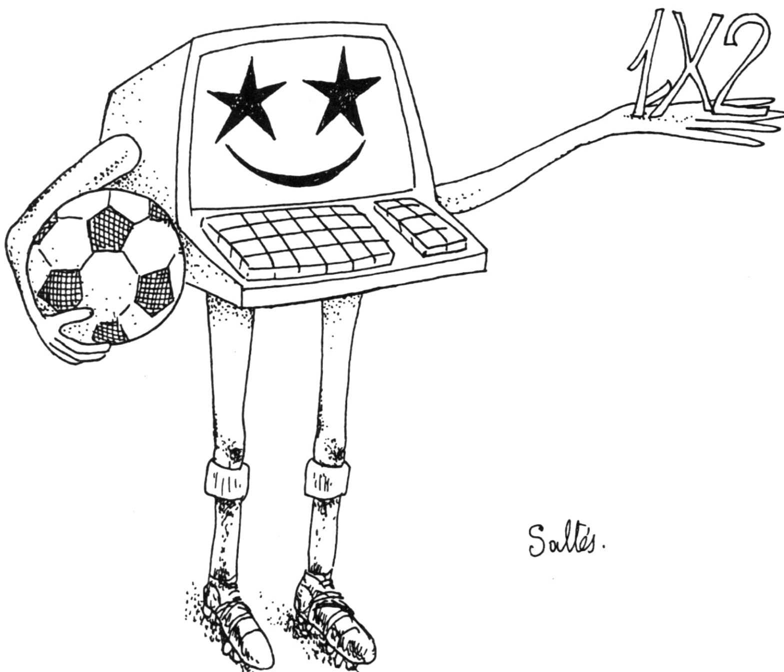
```
20 RANDOMIZE
25 LET A=INT(2*RND)
```

48. 1, X, 2 (12.4)

Prepara un programa para obtener al azar uno de los resultados 1, X, 2 de una quiniela.

Con la instrucción LET N=INT(3*RND(1)) conseguirás que tu ordenador escoja al azar uno de estos tres números: 0, 1, 2, y lo guarde en N. Los números 1 y 2 te valen para la quiniela, pero cuando la N es cero, el ordenador tendrá que escribir la letra X. Eso es precisamente lo que hace el programa siguiente:

```
   10 REM 1 X 2
(1)20 LET N=INT(3*RND(1))
   30 IF N=0 THEN PRINT "X":GOTO 50
   40 PRINT N
   50 END
```



(1) Si tu ordenador es un Spectrum, sustituye la línea 20 por estas dos:

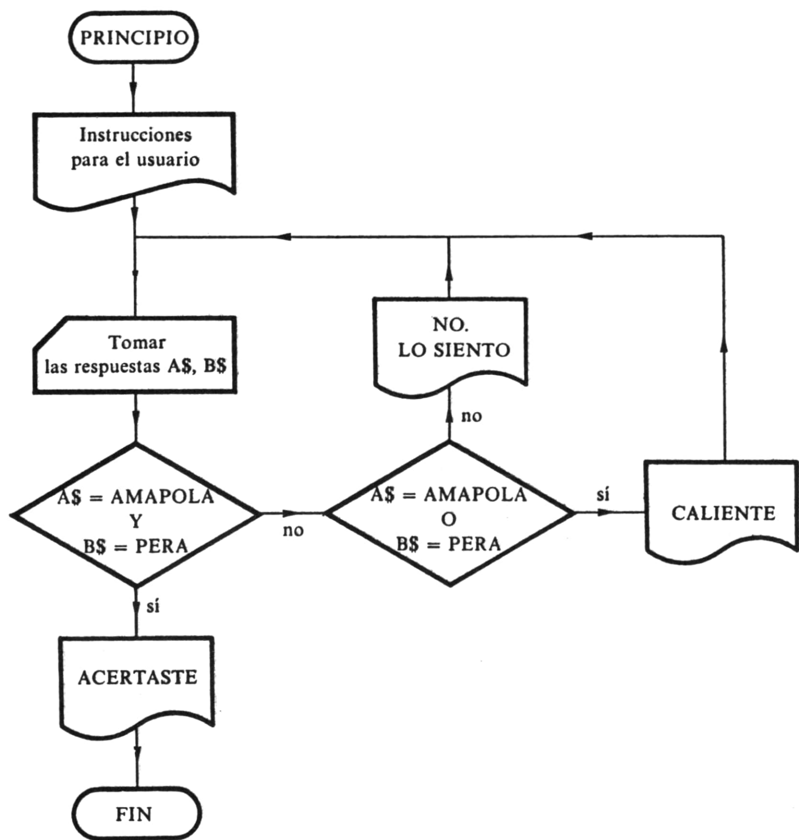
```
20 RANDOMIZE
25 LET N=INT(3*RND)
```

49. **ACERTIJO** (13.1)

El ordenador "piensa" una flor (AMAPOLA) y una fruta (PERA). Un amigo tuyo tiene que adivinarle los dos nombres. Prepara un programa para ello.

Vamos a dividir el programa en dos bloques: INSTRUCCIONES PARA EL USUARIO y CONTROL DE LAS RESPUESTAS.

Con el segundo haremos que tome las respuestas y que las controle. Si las respuestas son ambas correctas, dirá ACERTASTE y terminará el programa. Si una de las dos es buena, escribirá CALIENTE y dará una nueva oportunidad. Si ninguna de las dos respuestas es la correcta, escribirá NO. LO SIENTO y dará otra oportunidad.



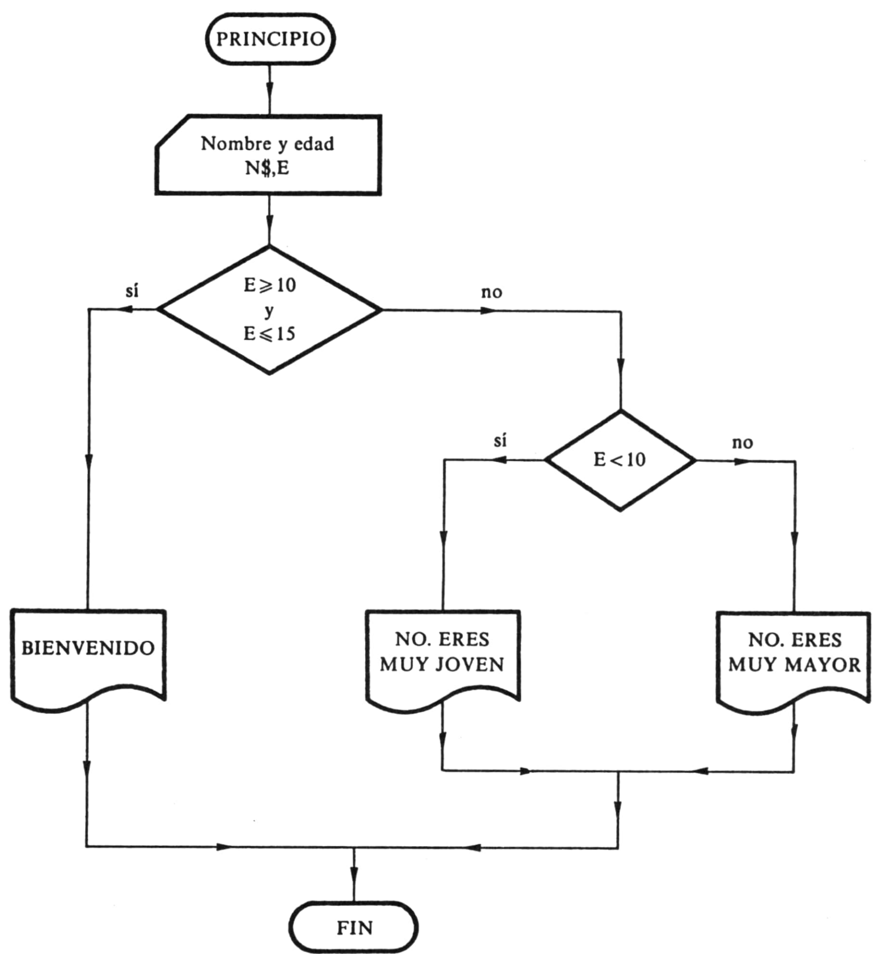
```
   10 REM ACERTIJO
   20 REM INSTRUCCIONES PARA EL USUARIO
(1)30 (BORRAR LA PANTALLA)
   40 PRINT "NO PEDIRE AL OLMO, AMOR,"
   50 PRINT "LO QUE NO ME PUEDE DAR"
   60 PRINT "(QUE VUELVAS, AMOR)."
   70 PRINT "YA NO GRABARE EN LA ARENA"
   80 PRINT "NI EN LAS ESPUMAS DEL MAR"
   90 PRINT "(AMOR QUE NO VOLVERA)"
   100 PRINT "(AMOR DE OLA EN PRIMAVERA)"
   110 PRINT "UNA FLOR QUE NUNCA SEA"
   120 PRINT "QUE VUELVAS, AMOR."
   130 PRINT:PRINT:PRINT
   140 PRINT "ESCRIBE LOS NOMBRES DE LA FLOR
       Y DE LA FRUTA SEPARADOS POR UNA COMA"
   200 REM CONTROL DE LAS RESPUESTAS
   210 INPUT A$,B$
   220 IF A$="AMAPOLA" AND B$="PERA" THEN PR
       INT "ACERTASTE":GOTO 250
   230 IF A$="AMAPOLA" OR B$="PERA" THEN PRI
       NT "CALIENTE":GOTO 210
   240 PRINT "NO. LO SIENTO.":GOTO 210
   250 END
```

(1) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT "(SHIFT) (CLR HOME)".

50. CLUB DE AMIGOS (13.2)

Se quiere formar el Club de Amigos de BASIC JUNIOR, para chicos de 10 a 15 años. Haz un programa para que el ordenador admita (o no) a los socios según su edad.

Puedes dividir el programa en dos partes. La primera, para preguntar el nombre y la edad de la persona; el nombre puedes meterlo en la variable de cadena N$ y la edad en la variable numérica E. La segunda parte, para controlar la edad y comunicar la decisión que corresponda por medio de *mensajes personales* (aprovechando que el nombre está en N$).



```
10 REM CLUB DE AMIGOS DE BASIC JUNIOR
20 REM TOMAR LOS DATOS
30 INPUT "NOMBRE";N$
40 INPUT "EDAD";E
50 REM CONTROLAR LA EDAD
60 IF E>=10 AND E<=15 THEN PRINT N$;". BI
   ENVENIDO AL CLUB":GOTO 90
70 IF E<10 THEN PRINT N$;". NO PUEDES ENT
   RAR. ERES MUY JOVEN":GOTO 90
80 PRINT "LO SENTIMOS ";N$;". NO PUEDES E
   NTRAR PORQUE ERES MUY MAYOR"
90 END
```

Observa que la línea 60 sirve para admitir al club únicamente a aquellos chicos cuya edad E sea, a la vez, mayor o igual que 10 y menor o igual que 15. Las dos líneas siguientes son para los no admitidos. La 70 para los que son muy pequeños y la 80 para los que son muy mayores.

51. EL ORDENADOR ENAMORADO (13.3)

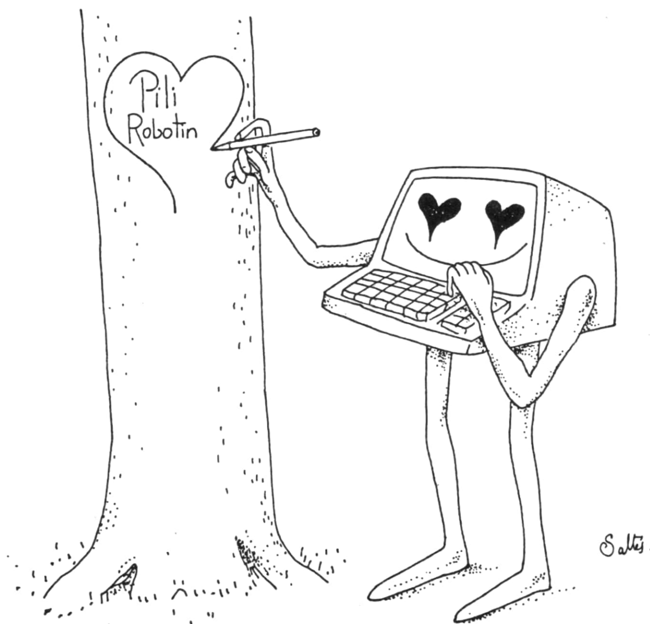
Los ordenadores también tienen su corazoncito. El tuyo está enamorado de Pilar. Prepárale un programa para que pregunte:

QUIEN ERES?

Y conteste TE AMO, cuando el nombre escrito sea PILAR o PILI. En otros casos que conteste con un frío HOLA.

```
10 REM ORDENADOR ENAMORADO
20 INPUT "QUIEN ERES";N$
30 IF N$="PILAR" OR N$="PILI" THEN PRINT
   "!!!  TE AMO  !!!":GOTO 50
40 PRINT "HOLA"
50 END
```

El programa es muy sencillo: la línea 20 escribe QUIEN ERES? y toma el nombre de la persona para guardarlo en la variable N$. La 30 comprueba si el nombre es PILAR o PILI, y en caso afirmativo escribe TE AMO. En caso contrario pasa a la línea siguiente para escribir HOLA.



52. CAJERA AUTOMATICA (13.4)

Hay rebajas en una tienda de tejidos. Venden los pares de calcetines a 150 ptas., las bufandas a 800 ptas. y los jerseys a 3.100 ptas. Si la compra es de más de 5.000 ptas., hacen un 20 por 100 de descuento. También lo hacen si te llevas 3 o más prendas. Debes saber que aunque te lleves muchas prendas, y tu factura sea de mucho dinero, el descuento seguirá siendo del 20 por 100, no del 40 por 100.

La cajera dispone de un ordenador en el que introduce el número de artículos de cada clase que has comprado, e inmediatamente aparece en la pantalla el precio total. Diseña un programa que haga esto.

Se puede dividir el programa en cuatro bloques para:

1.º Introducir los datos

2.º Calcular el total de pesetas y el número total de artículos

3.º Si corresponde, hacer el descuento

4.º Escribir el total a pagar

Puedes utilizar estas variables:

C : número de calcetines
B : número de bufandas
J : número de jerseys
T : total de pesetas
N : total de artículos

Es evidente que el *total de pesetas sin descuento* es:

```
T = C*150 + B*800 + J*3100
```

y que el número total de artículos es:

```
N = C + B + J
```

Si te correspondiera el 20 % de descuento, pagarías el 80 % de T. Para obtener este porcentaje basta calcular el producto 0.80*T; por eso el programa va a llevar la asignación:

```
LET T = 0.80*T
```

que da directamente el precio reducido y lo guarda *otra vez* en la variable T. Por eso, si $T > 5000$ (has comprado por un valor superior a 5000 pesetas) o $N \geqslant 3$ (llevas tres o más prendas), pagas el 80 %, como indica la línea:

```
IF T>5000 OR N>=3 THEN LET T = 0.8*T
```

El programa puede quedar así:

```
10 REM CAJERA AUTOMATICA
20 INPUT "CALCETINES";C
30 INPUT "BUFANDAS";B
40 INPUT "JERSEYS";J
50 LET T=C*150+B*800+J*3100
60 LET N=C+B+J
70 IF T>5000 OR N>=3 THEN LET T=.8*T
80 PRINT "TOTAL A PAGAR ";T;" PESETAS"
90 END
```

Observa que, en el caso de que no haya que hacer descuento, el valor de T, calculado en la línea 50, no se cambia y llega intacto a la línea 80, que lo escribe.

53. REFLEJOS (14.1)

Analiza este programa que sirve para medir tus reflejos. Inmediatamente después de ponerlo en marcha tienes que pulsar la tecla (*) .

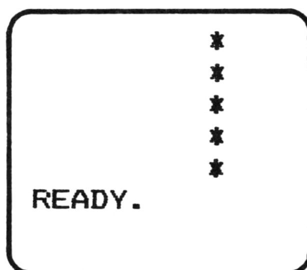
```
    10 REM REFLEJOS
(1) 20 (BORRAR LA PANTALLA)
    30 PRINT TAB(20);"*"
(2) 40 GET A$:IF A$<>"*" THEN GOTO 30
    50 END
```

La línea 20 borra la pantalla y lleva el cursor al rincón superior izquierdo.

La 30 escribe en la posición 20 de la siguiente línea de pantalla un asterísco (*).

La línea 40 toma el carácter de la tecla que hayas apretado en ese preciso instante y lo guarda en A$ (si no has pulsado tecla alguna, guarda en A$ la cadena vacía). Seguidamente comprueba si el contenido de A$ *no es* el asterisco (*). Si se verifica esta condición, vuelve a 30 para escribir *de nuevo* otro asterisco en la línea siguiente. Si no se cumple la condición (porque has apretado (*)), el programa pasa a la línea 50, donde termina.

Pon en marcha el programa escribiendo RUN y apretando (RETURN) . Cuanto antes pulses la tecla (*), menos asteriscos habrá en la pantalla. Esto mide tus reflejos.



(1) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT "(SHIFT) (CLR HOME)".

(2) Si tu ordenador es un Spectrum debes poner:

```
40 LET A$=INKEY$:IF A$<>"*" THEN GOTO 30
```

54. ESPERO A QUE ACABES (14.2)

Analiza primero, y ejecuta después, este programa:

```
   10 REM USO DEL GET
   20 PRINT "ESTOY ESPERANDO A QUE ACABES DE
      LEER ESTO"
   30 PRINT "PARA CONTINUAR PULSA UNA TECLA"
(1)40 GET A$:IF A$="" THEN GOTO 40
   50 PRINT "CONTINUO"
   60 END
```

En este programa se presenta una situación típica: el ordenador escribe un letrero en la pantalla (líneas 20 y 30) y detiene la ejecución del programa (sólo aparentemente) hasta que el usuario lo haya leído, momento en el que, pulsando una tecla cualquiera, la ejecución continúa.

La línea:

```
40 GET A$:IF A$= THEN GOTO 40
```

es muy interesante: Mientras no pulses una tecla, el ordenador está yendo de la línea 40 a la misma línea 40..., y el programa no avanza. Si aprietas una tecla cualquiera, el programa continúa por la línea siguiente. Vamos a explicarlo:

Mientras no pulses una tecla, la variable de cadena A$ estará vacía, *se cumplirá* la condición A$ = “”, y el programa “volverá” a la misma línea. Cuando hayas apretado una tecla, A$ no estará vacía y *no se cumplirá* la condición A$ = “”; por tanto el programa pasará a la siguiente línea.

---

(1) Si tu ordenador es un Spectrum debes poner:

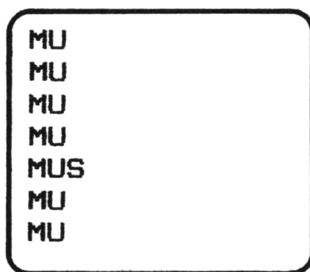
```
40 LET A$=INKEY$:IF A$="" THEN GOTO 40
```

55. GET (INKEY$) (14.3)

Pasa el programa siguiente y observa lo que ocurre cuando tocas una tecla cualquiera.

```
   10 REM GET
(1)20 GET A$
   30 IF A$="" THEN PRINT "MU":GOTO 20
   40 PRINT "MUS":GOTO 20
   50 END
```

Al pasar el programa y tocar una tecla, la pantalla queda así:



Al ejecutarse la línea 20, se guarda en la variable A$ el carácter de la tecla apretada en ese preciso instante; si no se aprieta ninguna tecla, entonces se guarda en A$ la *cadena vacía* ("" sin espacio entre las comillas). Así que, cuando el programa pasa a la línea 30, en A$ habrá una de estas dos cosas: *vacío* o un carácter (la letra P, por ejemplo).

Si A$ tiene la cadena vacía, entonces se cumple la condición A$ = "", el ordenador escribe MU y vuelve a la línea 20.

Si A$ tiene un carácter, entonces la condición A$ = "" no se cumple y el ordenador pasa a la línea siguiente (a la 40), que le manda escribir MUS y regresar a la línea 20. Aunque pulses rapidísimamente otra vez la tecla, no llegarás a tiempo de escribir MUS en la línea siguiente: ¡el ordenador *ya* ha pasado por la línea 20, ha cargado la cadena vacía y ha escrito MU!

---

(1) Si tu ordenador es un Spectrum, debes poner:

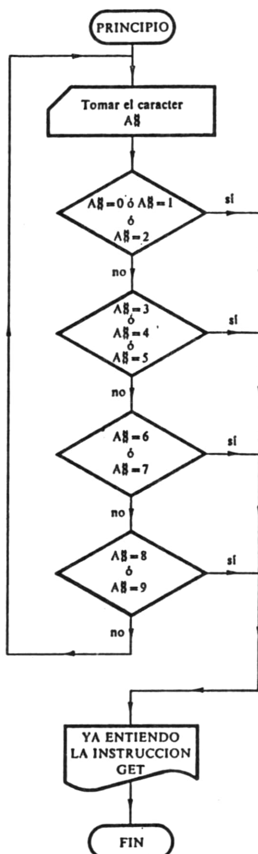
```
20 LET A$=INKEY$
```

56. YA ENTIENDO (14.4)

Haz un programa para que, dando a una tecla numérica, te presente un mensaje que diga: "YA ENTIENDO LA INSTRUCCION GET".

Mediante la instrucción GET, guardamos en la variable A$ el carácter de la tecla pulsada (o la cadena vacía, si no apretamos ninguna tecla).

A continuación se comprueba si A$ contiene un carácter numérico. Esto lo haremos en cuatro fases: primero se ve si A$ es 0, 1 ó 2, en cuyo caso el ordenador escribe el letrero. Si no, se comprueba si es 3, 4 ó 5. Y así sucesivamente, según muestra el diagrama de flujo.




```
10 REM YA ENTIENDO

(1)20 GET A$

30 IF A$="0" OR A$="1" OR
   A$="2" THEN GOTO 80

40 IF A$="3" OR A$="4" OR
   A$="5" THEN GOTO 80

50 IF A$="6" OR A$="7" TH
   EN GOTO 80

60 IF A$="8" OR A$="9" TH
   EN GOTO 80

70 GOTO 20

80 PRINT "YA ENTIENDO LA
   INSTRUCCION GET"

90 END
```

(1) Si tu ordenador es un Spectrum, debes poner:

```
20 LET A$=INKEY$
```

57. BANDERA (15.1)

Utiliza ciclos FOR-NEXT para hacer un programa que dibuje esta bandera:

```
X X X X X
X X X X X
X X X X X
X
X
X
X
```

Mira el dibujo y verás que se compone de la línea X X X X X, repetida tres veces, y de la línea X, repetida cuatro veces.

Este puede ser el algoritmo para dibujar la bandera:

1. Borrar la pantalla
2. Repetir tres veces
   Escribir X X X X X
3. Repetir cuatro veces
   Escribir X
4. Fin

Y éste el programa:

```
    10 REM BANDERA
(1) 20 (BORRAR LA PANTALLA)
   ┌30 FOR I=1 TO 3
   │40 PRINT "X X X X X"
   └50 NEXT I
   ┌60 FOR J=1 TO 4
   │70 PRINT "X"
   └80 NEXT J
    90 END
```

(1) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT "[SHIFT] [CLR HOME]".

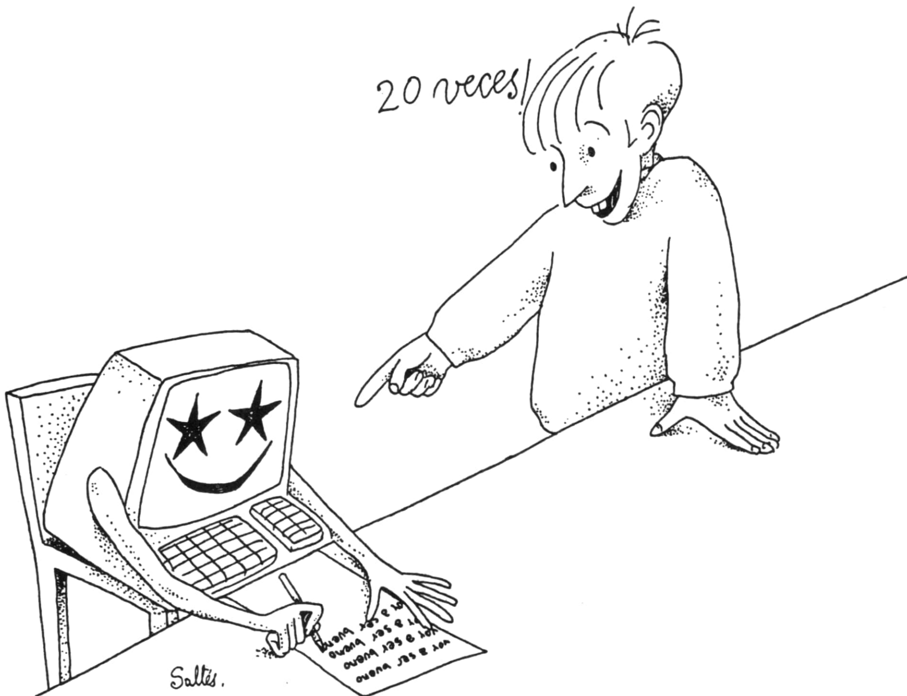
58. **CASTIGO** (15.2)

Castiga a tu ordenador a escribir veinte veces:

VOY A SER BUENO

El programa es muy sencillo:

```
 10 REM CASTIGO
┌20 FOR I=1 TO 20
│30 PRINT "VOY A SER BUENO"
└40 NEXT I
 50 END
```



59. MULTIPLOS DE 3 (15.3)

Haz un programa que escriba los 50 primeros múltiplos de 3.

Los cincuenta primeros múltiplos, empezando por el propio 3, son:

3, 6, 9, 12, ..., 150

Puedes imaginar que estos múltiplos son generados por una variable M, que inicialmente vale 3 y que va cambiando de tres en tres hasta llegar a 150. Esta idea da lugar al programa:

```
 10 REM MULTIPLOS DE 3
┌20 FOR M=3 TO 150 STEP 3
│30 PRINT M;" ";
└40 NEXT M
 50 END
```

También puedes ver dichos múltiplos así:

3*1, 3*2, 3*3, 3*4, ..., 3*50

generados por el producto 3*I, cuando I varía desde 1 hasta 50. Con esta nueva idea tienes:

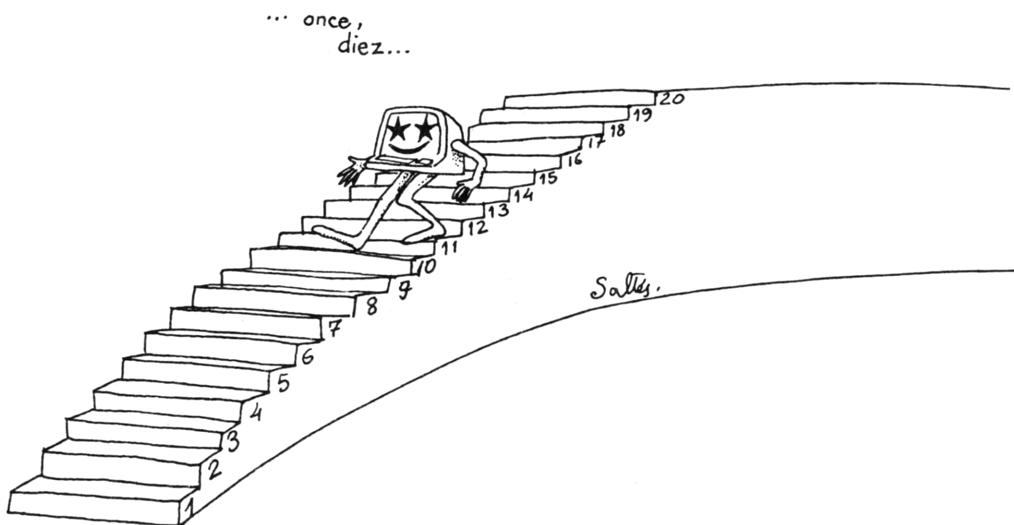
```
 10 REM MULTIPLOS DE 3
┌20 FOR I=1 TO 50
│30 PRINT 3*I;" ";
└40 NEXT I
 50 END
```

60. LA CUENTA ATRAS (15.4)

Escribe un programa que te presente en pantalla los números del 20 al 1. Observa que van disminuyendo de uno en uno. (El STEP puede tomar valores negativos).

Si utilizamos STEP -1, el programa queda así de sencillo:

```
 10 REM CUENTA ATRAS
┌20 FOR N=20 TO 1 STEP -1
│30 PRINT N
└40 NEXT N
 50 END
```



61. **PAUSA** (16.1)

La línea de programa:

```
FOR J=1 TO 5000:NEXT J
```

no hace nada, pero entretiene al ordenador recorriendo 5000 veces el ciclo FOR-NEXT, cosa que le lleva algún tiempo.

Este truco puedes emplearlo cuando necesites que la ejecución de un programa se detenga (aparentemente) unos instantes. Utilízalo para hacer un programa que:

1.º Borre la pantalla.
2.º Escriba "TE ESTOY DEJANDO TIEMPO PARA QUE LEAS ESTE LETRERO".
3.º Borre la pantalla.

```
   10 REM PAUSA
(1)20 (BORRAR LA PANTALLA)
   30 PRINT TAB(5);"TE ESTOY DEJANDO TIEMPO"
   40 PRINT
   50 PRINT TAB(5);"PARA QUE LEAS ESTE LETRE
      RO"
   60 FOR J=1 TO 5000:NEXT J
(1)70 (BORRAR LA PANTALLA)
   80 END
```

Como el letrero es un poco largo lo hemos repartido en dos líneas procurando que quede centrado.

(1) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT "(SHIFT)  CLR HOME".

62. EL RICO Y EL MENDIGO (16.2)

Un pobre le dijo a un rico: "Cada día, durante un mes, le pagaré 100.000 ptas.; a cambio Vd. me dará, el primer día una peseta, el segundo día dos pesetas, el tercero cuatro, es decir, cada día doble cantidad que la anterior, y así hasta acabar el mes".

Ante tan ventajosa propuesta, el rico aceptó inmediatamente, frotándose las manos.

Haz un programa que calcule las cantidades entregadas por cada uno, y escriba algún comentario jocoso al que se lo merezca.

Suponemos que el mes tiene treinta días. Las cantidades que entrega el pobre se acumulan en SP (Suma del Pobre). El valor de SP correspondiente a cada día es igual a la suma del valor del día anterior más las 100.000 ptas. de ese día, es decir:

```
LET SP = SP + 100000
```
(1)

El cálculo de las cantidades que da el rico es algo más difícil porque el dinero que entrega cada día no es constante. La cantidad diaria es la del día anterior multiplicada por dos. La variable DR (Diariamente por el Rico) recoge ese dinero.

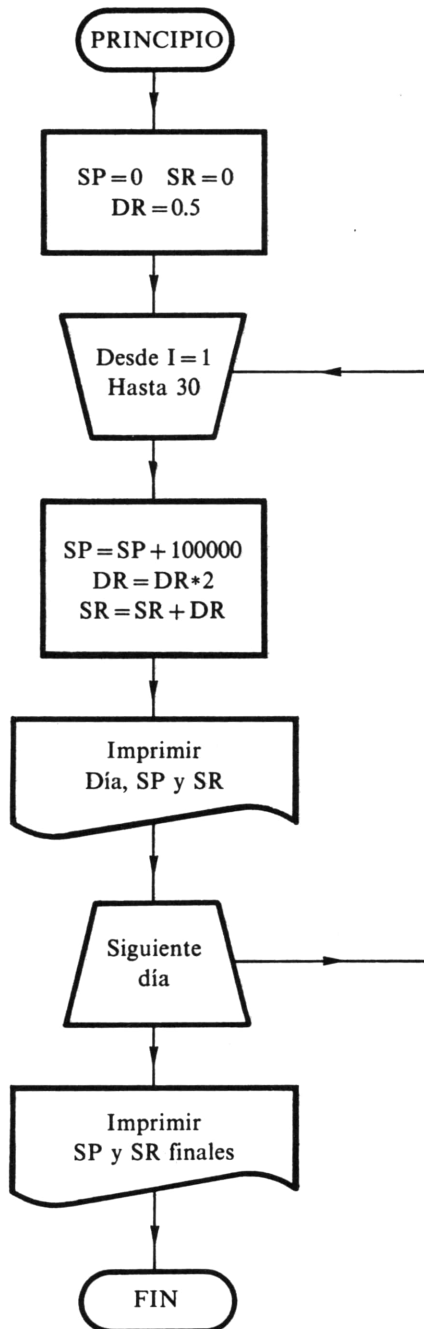
```
LET DR = DR*2
```
(2)

La suma del dinero que entrega el rico se guarda en SR. Para obtener esta suma en cada momento, hay que conocer la suma anterior y el dinero que ha entregado el rico ese día.

```
LET SR = SR + DR
```
(3)

Las instrucciones (1), (2) y (3) se incluyen en un ciclo FOR-NEXT que se repite 30 veces. Al final en SP está el dinero que entrega el pobre y en SR el que entrega el rico.

Si queremos saber los valores de SP y SR correspondientes a cada día del mes, escribiremos los números SP y SR a medida que se van calculando.


PRINCIPIO
SP = 0   SR = 0
DR = 0.5
Desde I = 1
Hasta 30
SP = SP + 100000
DR = DR*2
SR = SR + DR
Imprimir
Día, SP y SR
Siguiente
día
Imprimir
SP y SR finales
FIN

```
  10 REM EL POBRE Y EL RICO
(1)20 (BORRAR LA PANTALLA)
  30 LET SP=0:LET SR=0:LET DR=.5
  40 PRINT "CANTIDADES ENTREGADAS"
  50 PRINT
  60 PRINT "DIA";TAB(8);"POBRE";TAB(20);"RI
     CO"
  70 PRINT "========================="
  80 PRINT
  90 FOR I=1 TO 30
  100 LET SP=SP+100000
  110 LET DR=DR*2
  120 LET SR=SR+DR
  130 PRINT I;TAB(4);"---";TAB(7);SP;TAB(19
      );SR
  140 FOR J=1 TO 500:NEXT J
  150 NEXT I
  160 PRINT
  170 PRINT "EL POBRE ENTREGA:";SP
  180 PRINT
  190 PRINT "EL RICO ENTREGA:";SR
  200 PRINT
  210 PRINT "OBSERVA QUE EL DIA 22 EL RICO"
  220 PRINT "HA ENTREGADO YA MAS DINERO QUE
      EL POBRE"
  230 END
```

Si deseamos que el ordenador calcule el día en el que el rico ha entregado más dinero que el pobre, hay que intercalar dos líneas de control en el ciclo FOR-NEXT, por ejemplo entre 120 y 130.

```
122 IF SR>SP THEN LET K=K+1
124 IF K=1 THEN LET H=I
```

También hay que inicializar la variable K en la línea 30 (LET K=0).

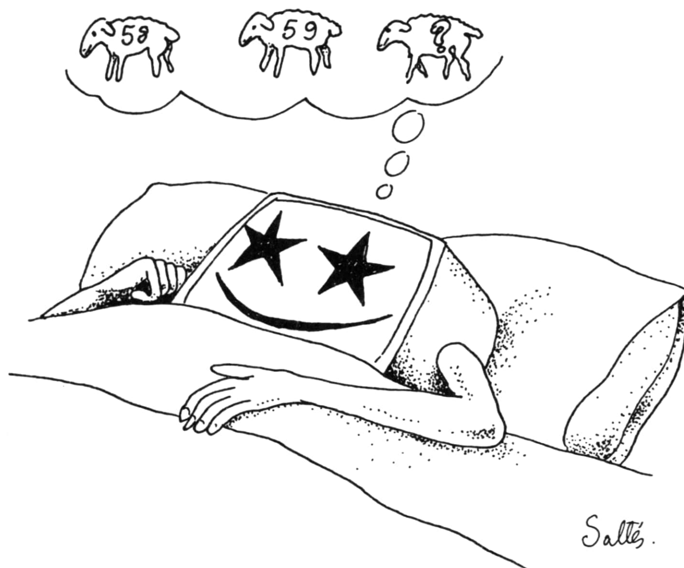
(1) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT "[SHIFT] [CLR HOME]".

63. SALIDA DE UN CICLO (16.3)

De un ciclo se puede salir sin necesidad de haberlo recorrido todas las veces.

```
10 REM SALIDA DE UN CICLO
20 PRINT "VOY A VER SI CUENTO HASTA 100"
30 FOR I=1 TO 100
40 PRINT I
50
60 NEXT I
70 PRINT "LO SIENTO: NO SE CONTAR MAS ALL
   A DE 59"
80 END
```

Completa la línea 50 para que el último número que se escriba sea el 59.



Los números se escriben en la pantalla con la instrucción de la línea 40. Escrito el número 59, la línea 50 tiene que establecer la condición de que si I = 59, entonces salte a la línea 70. Así se sale del ciclo y no se imprimen más números.

```
50 IF I=59 THEN GOTO 70
```

64. ENTRADA EN UN CICLO (16.4)

Observa lo que ocurre cuando ejecutas el siguiente programa:

```
10 FOR I=1 TO 5
20 PRINT I*I
30 NEXT I
40 PRINT "OBSERVA LO QUE PASA"
50 GOTO 20
60 END
```

Si ejecutamos el programa aparece en la pantalla:

```
 1
 4
 9
 16
 25
OBSERVA LO QUE PASA
 36
?NEXT WITHOUT FOR ERROR IN 30
```

El ciclo FOR-NEXT imprime los cuadrados de los cinco primeros números. A la salida del ciclo, I vale 6.

Después de imprimir "OBSERVA LO QUE PASA" la instrucción 50 envía a la línea 20, y como I vale 6, se imprime su cuadrado 36. A continuación aparece un mensaje de error que quiere decir algo así como:

ERROR EN 30 POR HABER LLEGADO A NEXT SIN PASAR POR FOR

Es decir, no se acepta llegar a la instrucción NEXT sin haber pasado previamente por la instrucción FOR.

Si en vez de poner GOTO 20 ponemos GOTO 10, entramos en el ciclo por la primera instrucción y no hay error, pero si la entrada es por cualquier otra línea cometemos siempre error.

65. DISTANCIAS (17.1)

Haz un programa que escriba diez capitales europeas y su distancia a Madrid.

El programa está dividido en dos bloques:

El primero, contiene en líneas DATA las capitales y las distancias correspondientes.

El segundo bloque consiste en un ciclo FOR-NEXT que realiza:

a) *Lectura de la capital y de la distancia:* En C$ se guarda el nombre de la capital y en D la distancia.

b) *Escritura de estos datos.*

Date cuenta de que para I=1, C$ contiene BERLIN y D guarda 2527, para I=2, BERNA sustituye a BERLIN y 1528 a 2527 y así sucesivamente.

```
   10 REM DISTANCIAS A MADRID
(1)20 (BORRAR LA PANTALLA)
   30 DATA "BERLIN",2527,"BERNA",1528
   40 DATA "BRUSELAS",1553,"BUDAPEST",2634
   50 DATA "LISBOA",648,"ESTOCOLMO",3367
   60 DATA "PARIS",1261,"PRAGA",2391
   70 DATA "ROMA",2288,"VIENA",2377
   80 FOR I=1 TO 10
   90 READ C$,D
   100 PRINT C$,D
   110 NEXT I
   120 END
```

(1) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT "SHIFT  CLR HOME".

66. TABLA DE NUMEROS (17.2)

Preapara un programa que escriba la tabla:

```
1  UNO
2  DOS
3  TRES
.  .
.  .
.  .
10 DIEZ
```

Te sugerimos que emplees estas líneas:

```
100 DATA "UNO","DOS","TRES","CUATRO","CINCO"
110 DATA "SEIS","SIETE","OCHO","NUEVE","DIEZ"
```

y que las leas con un ciclo FOR-NEXT, con el que también podrás escribir la tabla.

El ciclo FOR-NEXT (30-60) cumple dos tareas. La primera, cargar en la variable N$ el nombre del número, tomándolo de las líneas DATA, y la segunda imprimir el número, y a su derecha el nombre.

```
   10 REM TABLA DE NUMEROS
(1)20 (BORRAR LA PANTALLA)
   30 FOR I=1 TO 10
   40 READ N$
   50 PRINT I;TAB(5);N$
   60 NEXT I
(2)100 DATA "UNO","DOS","TRES","CUATRO","CIN
       CO"
(2)110 DATA "SEIS","SIETE","OCHO","NUEVE","D
       IEZ"
   120 END
```

Date cuenta de que para I = 1, N$ toma el valor UNO; si I = 2 el valor UNO es sustituido por DOS, y así sucesivamente.

---

(1) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT "[SHIFT] [CLR HOME]".

(2) Hay ordenadores en los que puedes prescindir de las comillas.

67. TABLAS DE SUMAR (17.3)

Programa la tabla de sumar del 5. Modifica el programa para que salga la tabla de cualquier número que tú le des.

El ordenador tiene que escribir:

```
5 + 0  = 5
5 + 1  = 6
5 + 2  = 7
.     .    .
.     .    .
.     .    .

5 + 9  = 14
```

lo cual se hace cómodamente con un ciclo FOR-NEXT, que escribirá para cada valor de su índice I, desde 0 a 9, lo siguiente:

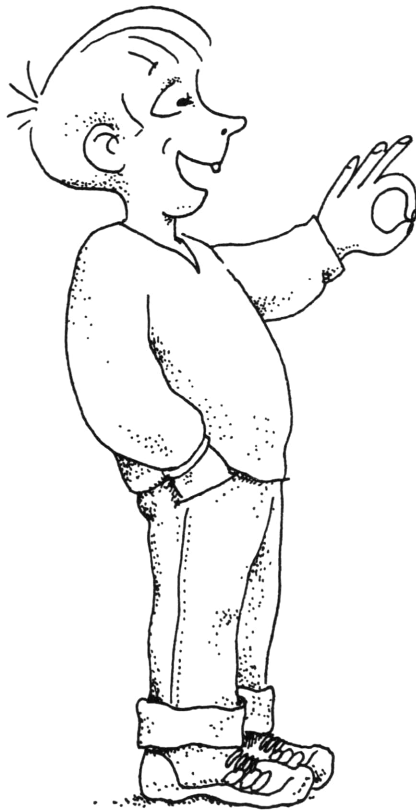
El número 5
El signo más (+)
El valor actual de I
El signo igual (=)
El resultado de la suma 5+I

Tanto el signo más como el igual deben estar entrecomillados; en caso contrario habría error.

```
10 REM TABLA DE SUMAR DEL 5
20 FOR I=0 TO 9
30 PRINT 5;" +";I;" =";5+I
40 NEXT I
50 END
```

Ahora vamos a modificar el programa para que nos dé la tabla de sumar de cualquier número. Una instrucción INPUT antes del ciclo, permite guardar el número que queramos en la variable N. Al no saber previamente de qué número se trata, la instrucción 30 se ha de modificar poniendo en lugar de 5, la variable N.

```
10 REM TABLA DE SUMAR
15 INPUT "QUE NUMERO QUIERES REPASAR";N
20 FOR I=0 TO 9
30 PRINT N;" +";I;" =";N+I
40 NEXT I
50 END
```





68. CALCULO DE LA MEDIA (17.4)

Haz un programa para calcular la media de una serie de datos. Imagínate que los ocho compañeros de un equipo de trabajo decidieran hacer un fondo común con las notas de matemáticas y después repartirse el fondo a partes iguales. El número que correspondería a cada uno sería la nota media. Supongamos que las notas fueran:

9, 7, 3, 6, 9, 5, 2 y 9

La suma de las notas es $S=50$.

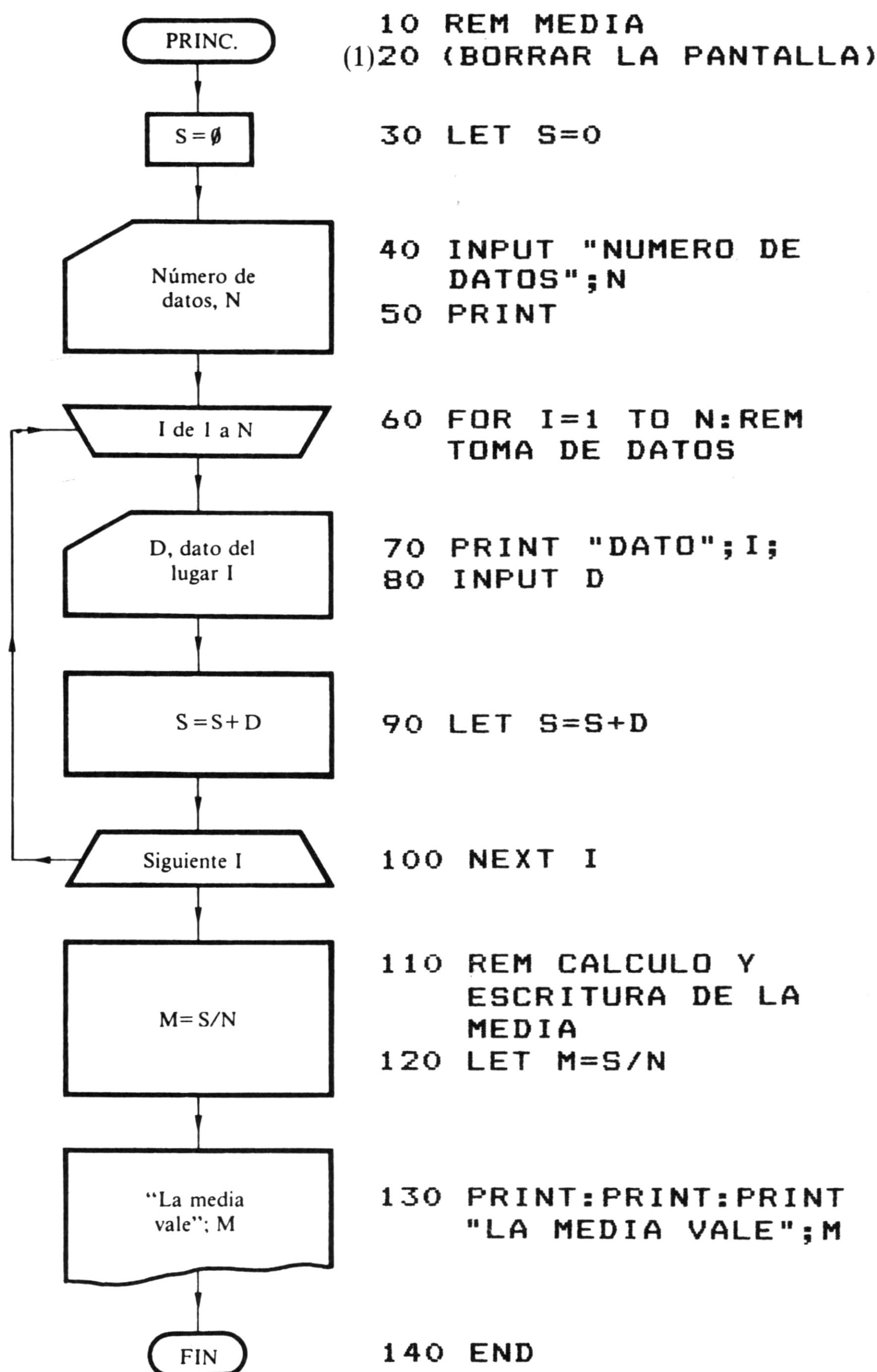
La nota media es

$$M = \frac{50}{8} = 6{,}25$$

En general, la media de una serie de datos es:

$$\text{Media} = \frac{\text{Suma de todos los datos}}{\text{Número de datos}}$$

El programa que tienes que diseñar ha de valer para cualquier número de datos, no exclusivamente para el caso de ocho, como en el ejemplo. El número de datos puedes meterlo en la variable N con una instrucción INPUT. Los datos los metes uno a uno en la variable D, que los acumula en la variable S, en donde van quedando las sumas parciales. Al final, en S estará la suma total. El siguiente diagrama te ayudará a construir el programa:




```
   10 REM MEDIA
(1)20 (BORRAR LA PANTALLA)
   30 LET S=0
   40 INPUT "NUMERO DE
      DATOS";N
   50 PRINT
   60 FOR I=1 TO N:REM
      TOMA DE DATOS
   70 PRINT "DATO";I;
   80 INPUT D
   90 LET S=S+D
  100 NEXT I
  110 REM CALCULO Y
      ESCRITURA DE LA
      MEDIA
  120 LET M=S/N
  130 PRINT:PRINT:PRINT
      "LA MEDIA VALE";M
  140 END
```

(1) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT "SHIFT CLR HOME".

69. **RULETA** (18.1)

Forma un número de seis cifras haciendo girar una ruleta de 0 a 9.

```
   10 REM RULETA
(1)20 (BORRAR LA PANTALLA)
   30 LET N=0
   40 FOR I=1 TO 6
(2)50 LET C=INT(10*RND(1))
   60 IF C=0 AND I=1 THEN GOTO 50
   70 LET N=N*10+C
   80 NEXT I
   90 PRINT "EL NUMERO QUE HA SALIDO ES:";N
   100 END
```

Con el ciclo FOR-NEXT "giramos la ruleta" 6 veces, para sacar sucesivamente las cifras C del número N. A medida que van saliendo las cifras, las vamos montando en la variable N, como muestra el siguiente ejemplo:

Supongamos que se obtienen sucesivamente las cifras 3,5,7,2,3 y 1. El número N se va formando de este modo:

```
N = 10 * 0 + 3 =  3
N = 10 * 3 + 5 =  35
N = 10 * 35 + 7 =  357
N = 10 * 357 + 2 =  3572
N = 10 * 3572 + 3 =  35723
N = 10 * 35723 + 1 =  357231
```

Así pues, el número obtenido es el 357231. En la línea 60 hemos puesto un control para evitar que la primera cifra obtenida sea 0 en cuyo caso el número no tendría seis cifras.

---

(1) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT "(SHIFT) (CLR HOME)".

(2) Si tu ordenador es un Spectrum, sustituye la línea 50 por estas dos:

```
15 RANDOMIZE
50 LET C=INT(10*RND)
```

70. LANZAMIENTO DE UNA MONEDA (18.2)

Haz un programa que imite el lanzamiento de una moneda 100 veces, y que indique al final el número de caras y de cruces obtenido.

Imitamos el lanzamiento de una moneda con la función RND. Sabemos que:

```
0 < RND(1) < 1
```

Si multiplicamos por dos y tomamos la parte entera, obtenemos uno de estos dos números, 0 ó 1. Si sale cero, será cara y si sale uno, será cruz.

El número de caras se guarda en la variable CA y el de cruces en la variable CR.

Los cien números aleatorios se generan con un ciclo FOR-NEXT.

```
   10 REM LANZAMIENTO DE UNA MONEDA
(1)20 (BORRAR LA PANTALLA)
   30 PRINT "LANZAMIENTO DE UNA MONEDA 100 V
      ECES"
   40 PRINT:PRINT
   50 LET CA=0:LET CR=0
   60 FOR I=1 TO 100
(2)70 LET A=INT(RND(1)*2)
   80 IF A=0 THEN LET CA=CA+1:GOTO 100
   90 LET CR=CR+1
   100 NEXT I
   110 PRINT "HAN SALIDO";CA;"CARAS Y";CR;"C
       RUCES"
   120 END
```

---

(1) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT "[SHIFT] [CLR HOME]".

(2) Si tu ordenador es un Spectrum, sustituye la línea 70 por estas dos:

```
15 RANDOMIZE
70 LET A=INT(RND*2)
```

71. JUEGO DE DADOS (18.3)

El juego consiste en lanzar dos dados. Si la suma es 7 u 11, gana el ordenador, y si no, ganas tú. Programa una partida de diez juegos.

Recuerda que en el problema 45 simulamos el dado con la función RND. Para obtener un punto de un dado la instrucción es:

```
INT(RND(1)*6)+1
```

Un ciclo FOR-NEXT repite el juego diez veces. En D1 y D2 se guardan los números que han salido en los dados, y en S la suma de esos números.

El número de partidas ganadas por la máquina se recoge en MA y las ganadas por el jugador en JU.

En cada juego se imprimen los puntos obtenidos con los dados y el ganador.

Entre juego y juego hemos puesto una pausa para hacer más lento el desarrollo de la partida.

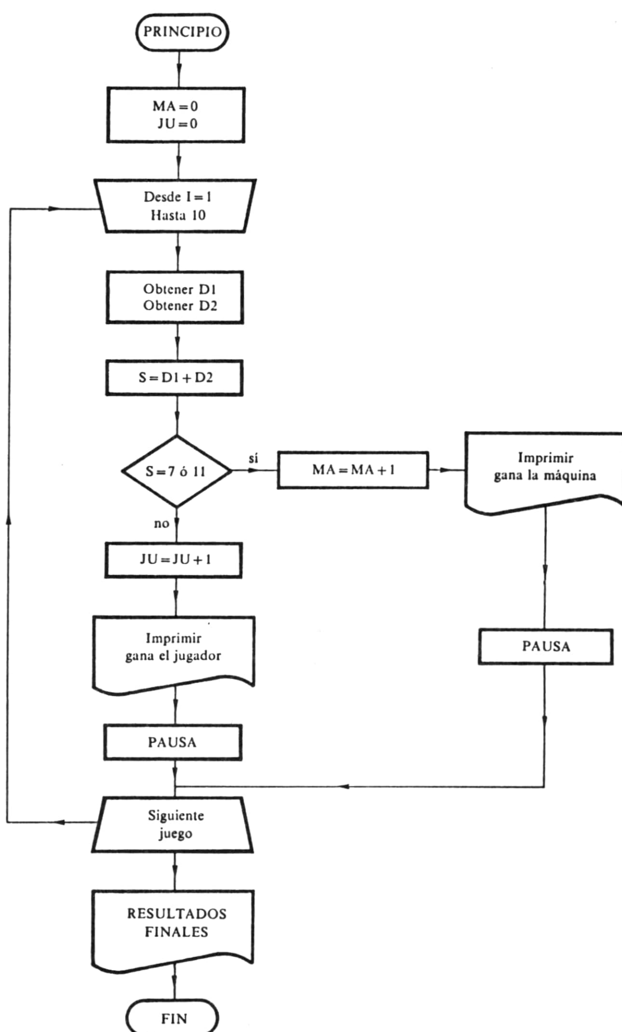
```
    10 REM JUGAR A LOS DADOS
(1) 20 (BORRAR LA PANTALLA)
    30 LET MA=0:LET JU=0
    40 PRINT "DADO 1";TAB(10);"DADO 2";TAB(20
       );"GANADOR"
    50 FOR I=1 TO 26:PRINT "-";:NEXT I
    60 PRINT
    70 FOR I=1 TO 10
(2) 80 LET D1=INT(RND(1)*6)+1
(2) 90 LET D2=INT(RND(1)*6)+1
    100 LET S=D1+D2
    110 IF S=7 OR S=11 THEN GOTO 160
    120 LET JU=JU+1
```

(1) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT "SHIFT CLR HOME".

(2) Si tu ordenador es un Spectrum, sustituye las líneas 80 y 90 por las siguientes:

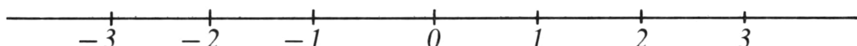
```
35 RANDOMIZE
80 LET D1=INT(RND*6)+1
90 LET D2=INT(RND*6)+1
```

```
130 PRINT TAB(2);D1;TAB(12);D2;TAB(20);"J
    UGADOR"
140 FOR J=1 TO 1000:NEXT J
150 GOTO 190
160 LET MA=MA+1
170 PRINT TAB(2);D1;TAB(12);D2;TAB(20);"M
    AQUINA"
180 FOR J=1 TO 1000:NEXT J
190 NEXT I
200 PRINT:PRINT "TU HAS GANADO";JU;"JUEGO
    S"
210 PRINT:PRINT "YO HE GANADO";MA;"JUEGOS
    "
220 END
```



72. EL SALTO DE LA PULGA (18.4)

Tienes una recta en la que están representados los números 0, 1, 2, 3, ... y −1, −2, −3, ... Imagina, como puedas, que en el punto 0 está situada una pulga que va a iniciar una serie de saltos de longitud uno, y que, en cada punto, decide al azar (como si lo echara a cara o cruz) si va a saltar hacia la derecha o hacia la izquierda.



Supón que la pulga da mil saltos y se detiene. ¿Cuál será la posición final? ¿Hasta qué puntos extremos (por la derecha y por la izquierda) habrá llegado en su loco paseo?

Te damos algunas pistas para que te ayuden a construir el programa. La variable P marca la posición de la pulga. Inicialmente P=0. Las posiciones sucesivas se establecen con la instrucción P=P+S, en donde S es el salto, que puede valer 1 ó −1, según sea hacia la derecha o hacia la izquierda. El salto se puede calcular así:

```
S = 2*X - 1
```

donde X es un número aleatorio, 0 ó 1:

```
X = INT(2*RND(1))
```

La posición más a la derecha la puedes guardar en la variable D. Al principio D=0. En D se llevan los "records" por la derecha, por lo que cambiará de valor cada vez que sea P>D. Será necesaria una línea del tipo:

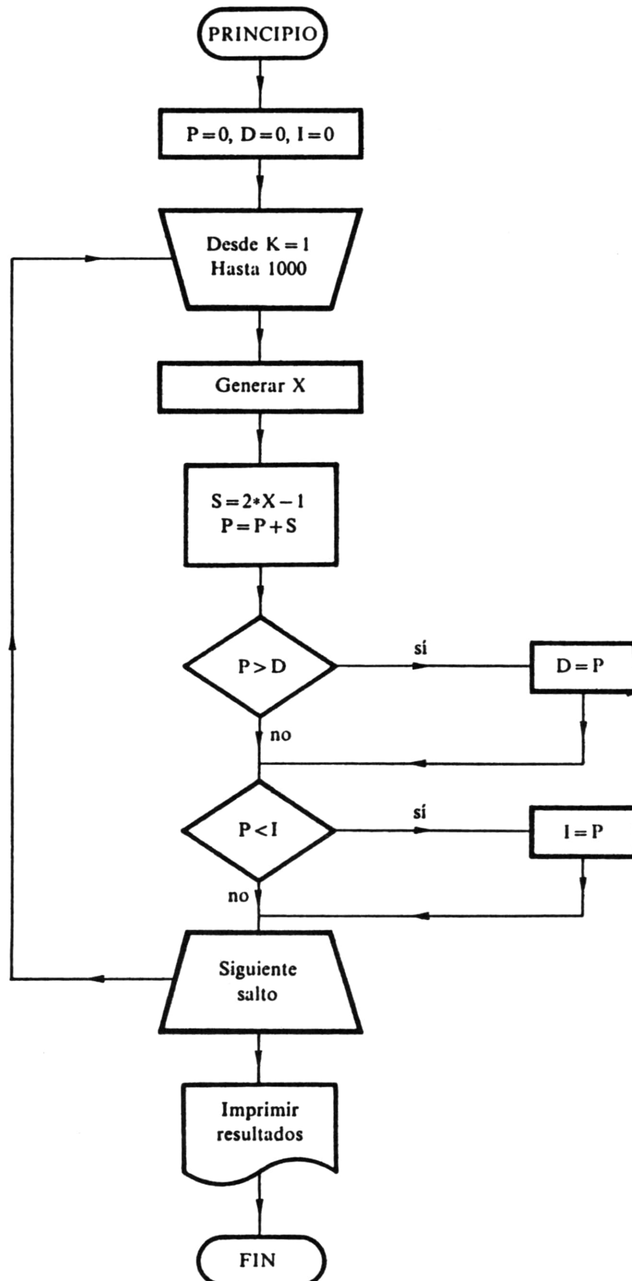
```
IF P>D THEN LET D=P
```

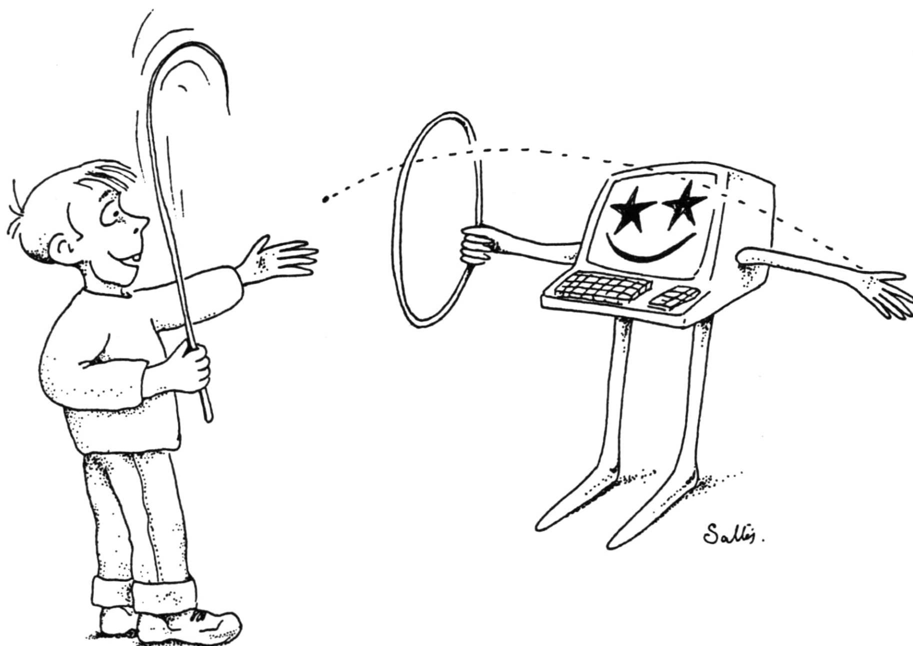
Análogas consideraciones podríamos hacer para la variable I, que marca la posición extrema izquierda.

El núcleo del programa está incluido en un ciclo FOR-NEXT, dentro del cual se realizan las siguientes acciones:

Obtener X (que será 0 ó 1)
Calcular S y P
Si P > D entonces hacer D = P
Si P < I entonces hacer I = P

Finalizado el ciclo se imprimen los resultados.





```
   10 REM SALTO DE LA PULGA
   20 LET P=0:LET D=0:LET I=0
   30 FOR K=1 TO 1000
(1)40 LET X=INT(RND(1)*2)
   50 LET S=2*X-1
   60 LET P=P+S
   70 IF P>D THEN LET D=P
   80 IF P<I THEN LET I=P
   90 NEXT K
   100 PRINT "LA POSICION FINAL ES";P
   110 PRINT "POR LA DERECHA HA LLEGADO HAST
       A";D
   120 PRINT "POR LA IZQUIERDA HA LLEGADO HA
       STA";I
   130 END
```

No se debe usar la variable I como índice del ciclo por estar reservada para la posición más a la izquierda. En caso de que se quiera utilizar para el ciclo, han de emplearse para las posiciones extremas otras variables. Por ejemplo, DE (posición más a la derecha) e IZ (posición más a la izquierda).

---

(1) Si tu ordenador es un Spectrum, sustituye la línea 40 por estas dos:

```
25 RANDOMIZE
40 LET X=INT(RND*2)
```

73. RECTANGULO DE ASTERISCOS (19.1)

Utiliza dos ciclos encajados para dibujar este rectángulo:

```
* * * * * * * * * *

* * * * * * * * * *

* * * * * * * * * *

* * * * * * * * * *
```

Para dibujar el rectángulo con 4 líneas horizontales de 10 asteriscos cada una, necesitamos dos ciclos encajados.

El ciclo interno dibuja, para cada valor de I, los diez asteriscos de la línea. Es decir, si I = 1, J varía de 1 a 10 y se imprimen los asteriscos de la primera horizontal; después, para I = 2, J varía nuevamente de 1 a 10 y se dibujan los asteriscos de la segunda línea y así sucesivamente.

```
   10 REM RECTANGULO DE ASTERISCOS
(1)20 (BORRAR LA PANTALLA)
   30 FOR I=1 TO 4
   40 FOR J=1 TO 10
   50 PRINT "* ";
   60 NEXT J
   70 PRINT
   80 NEXT I
   90 END
```

Fíjate que es necesario poner una instrucción PRINT en la línea 70 para que se realice el salto de una fila a otra. Si no se pusiera, el primer asterisco de una fila saldría al lado del último de la fila anterior y se estropearía el dibujo. Ocurre así, porque el punto y coma (;) de la instrucción 50 ordena que la impresión de cada asterisco se haga a la derecha del anterior y pegado a él. Observa también, que para evitar que salgan pegados los asteriscos hay que imprimir "* ".

(1) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT "(SHIFT) (CLR HOME)".

74. TABLA DE NUMEROS (19.2)

Con dos ciclos convenientemente encajados puedes representar este cuadro numérico:

```
0  1  2  3  4  5  6  7  8  9
0  1  2  3  4  5  6  7  8  9
0  1  2  3  4  5  6  7  8  9
0  1  2  3  4  5  6  7  8  9
```

Realizamos la impresión por filas. El ciclo interno escribe los diez números y el externo controla las filas. Para I = 1, se ejecuta el ciclo interior y se escriben los números de la primera fila; para I = 2, se ejecuta nuevamente el ciclo interno y se imprime la segunda fila, y así sucesivamente.

Es necesario introducir la línea 70:

```
70 PRINT
```

para que después de imprimir una fila, el cursor salte a la línea siguiente. Para que los números salgan separados usamos la instrucción: PRINT TAB (3*J + 1);J;. De este modo las impresiones se realizan en las columnas 1, 4, 7, 10, ..., 28.

```
   10 REM TABLA DE NUMEROS
(1)20 (BORRAR LA PANTALLA)
   30 FOR I=1 TO 4
   40 FOR J=0 TO 9
   50 PRINT TAB(3*J+1);J;
   60 NEXT J
   70 PRINT
   80 NEXT I
   90 END
```

(1) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT "[SHIFT] [CLR HOME]".

75. TABLAS DE MULTIPLICAR (19.3)

Con dos ciclos encajados escribe dos tablas de multiplicar; por ejemplo, la del 7 y la del 8. Modifica adecuadamente el programa anterior para escribir la del 7 y la del 9.

Vamos a escribir las tablas de modo que el cursor se mueva imprimiendo la tabla del 7 y del 8 simultáneamente.

Para I=1	7 x 1 = 7	8 x 1 = 8
Para I=2	7 x 2 = 14	8 x 2 = 16
.	.	.
.	.	.
.	.	.
Para I=10	7 x 10= 70	8 x 10= 80

Cada fila se imprime con el ciclo interno, cuya variable toma los valores 7 y 8. El control de las filas se realiza con el ciclo externo. En la línea 70 hemos utilizado la coma (,) para separar las dos tablas.

```
    10 REM TABLAS DE MULTIPLICAR
(1) 20 (BORRAR LA PANTALLA)
    30 PRINT "TABLA DEL 7","TABLA DEL 8"
    40 PRINT "-----------","-----------"
  ┌─50 FOR I=1 TO 10
  │┌60 FOR J=7 TO 8
  ││70 PRINT J;"x";I;"=";J*I,
  │└80 NEXT J
  │ 90 PRINT
  └─100 NEXT I
    110 END
```

Si en la línea 60 pones: 60 FOR J=7 TO 9 STEP 2 obtienes las tablas del 7 y del 9.

---

(1) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT "(SHIFT) (CLR HOME)".

76. SOPA DE LETRAS (19.4)

Como sabemos que te gusta la sopa, en las líneas DATA te damos los ingredientes para que cocines una buena sopa de letras.

```
200 DATA "A","A","Z","E","R","E","C","J",
    "I","B"
210 DATA "A","D","E","O","N","D","I","D",
    "P","A"
220 DATA "L","A","N","O","L","E","M","U",
    "R","A"
230 DATA "N","H","R","V","R","A","S","O",
    "Z","A"
240 DATA "T","M","N","I","S","J","A","L",
    "I"
```

Con dos ciclos anidados lee y escribe las letras formando un cuadro de 7 × 7. En él has de encontrar seis frutas.

El ciclo externo fija el número de la fila y el interno lee y escribe todas las letras de esa fila.

I \ J	1	2	3	4	5	6	7
1	A	A	Z	E	R	E	C
2	J	I	B	......	......	......	......

Hemos utilizado la función TAB, para que las letras se impriman en cada fila dejando un hueco entre ellas. Si ponemos punto y coma (;) salen juntas y se lee peor.

```
   10 REM SOPA DE LETRAS
(1)20 (BORRAR LA PANTALLA)
   30 FOR I=1 TO 7
   40 FOR J=1 TO 7
   50 READ L$
   60 PRINT TAB(2*J);L$;
   70 NEXT J
   80 PRINT
   90 NEXT I
   200 DATA "A","A","Z","E","R","E","C","J",
       "I","B"
   210 DATA "A","D","E","O","N","D","I","D",
       "P","A"
   220 DATA "L","A","N","O","L","E","M","U",
       "R","A"
   230 DATA "N","H","R","V","R","A","S","O",
       "Z","A"
   240 DATA "T","M","N","I","S","J","A","L",
       "I"
   250 END
```

(1) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT "(SHIFT) (CLR HOME)".

77. ESCRITURA VERTICAL (20.1)

Prepara un programa que, mediante la función MID$, escriba la palabra PAPEL en vertical, tanto de arriba abajo, como de abajo hacia arriba.

La palabra PAPEL la guardamos en P$ y con un ciclo FOR-NEXT *recorremos la palabra.*

Para imprimir de arriba abajo empleamos la instrucción MID$. Recuerda que MID$(P$,I,1) toma de la cadena P$ un carácter, a partir de la posición I.

En nuestro caso, I varía de 1 a 5.

Si I = 1 MID$(P$,1,1) = P

Si I = 2 MID$(P$,2,1) = A

Si I = 3 MID$(P$,3,1) = P

Si I = 4 MID$(P$,4,1) = E

Si I = 5 MID$(P$,5,1) = L

Para que la palabra se escriba al revés, sirve la misma instrucción, pero ahora hay que tomar las letras desde el final. Como PAPEL tiene 5 letras, la instrucción adecuada es:

MID$(P$,5+1-I,1)

De este modo:

Si I = 1 MID$(P$,5+1-1,1) = L

Si I = 2 MID$(P$,5+1-2,1) = E

Si I = 3 MID$(P$,5+1-3,1) = P

Si I = 4 MID$(P$,5+1-4,1) = A

Si I = 5 MID$(P$,5+1-5,1) = P

```
   10 REM ESCRITURA VERTICAL
(1)20 (BORRAR LA PANTALLA)
   30 LET P$="PAPEL"
   40 FOR I=1 TO 5
(2)50 PRINT MID$(P$,I,1),
(2)60 PRINT MID$(P$,5+1-I,1)
   70 PRINT
   80 NEXT I
   90 END
```

Hay una coma (,) al final de la línea 50 para que las dos palabras queden separadas. La línea 70 deja un espacio entre las letras de cada palabra.

Si queremos un programa que valga para cualquier palabra, hay que modificar ligeramente el anterior. En lugar de la línea 30 ponemos esta otra:

```
30 INPUT "CUAL ES LA PALABRA";P$
```

Además hay que calcular la longitud de la palabra; es decir, su número de letras. Esto se hace con la función LEN.

```
LET L=LEN(P$)
```

De este modo se carga en L el número de letras que contiene P$.

El programa queda así:

```
   10 REM ESCRITURA VERTICAL
(1)20 (BORRAR LA PANTALLA)
   30 INPUT "CUAL ES LA PALABRA";P$
   40 LET L=LEN(P$)
   50 FOR I=1 TO L
(2)60 PRINT MID$(P$,I,1),
(2)70 PRINT MID$(P$,L+1-I,1)
   80 PRINT
   90 NEXT I
   100 END
```

(1) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT "SHIFT CLR HOME".

(2) Para el Spectrum cambia esas líneas por éstas:

```
50 PRINT P$(I TO I)          60 PRINT P$(I TO I)
60 PRINT P$(5+1-I TO         70 PRINT P$(L+1-I TO
   5+1-I)                       L+1-I)
```

78. ATIRAGRAM (20.2)

Prepara un programa que, mediante la función MID$, escriba la palabra MARGARITA al revés.

El ejercicio es parecido al anterior. La palabra MARGARITA se carga en M$ y con un ciclo FOR-NEXT, recorremos la palabra de derecha a izquierda.

Para imprimir las letras empezando por la última, es necesaria la instrucción:

```
PRINT MID$(M$,I,1)
```

En nuestro caso I varía de 9 a 1:

```
I = 9   MID$(M$,9,1) = A

I = 8   MID$(M$,8,1) = T

.................................

I = 1   MID$(M$,1,1) = M
```

```
   10 REM MARGARITA
(1)20 (BORRAR LA PANTALLA)
   30 LET M$="MARGARITA"
   40 FOR I=9 TO 1 STEP -1
(2)50 PRINT MID$(M$,I,1);
   60 NEXT I
   70 END
```

(1) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT "[SHIFT] [CLR HOME]".

(2) Si tu ordenador es un Spectrum, sustituye la línea 50 por ésta:

```
PRINT M$(I TO I);
```

79. PLURALES (20.3)

Prepara un programa que escriba el plural de las palabras terminadas en Z, como capaz, nuez, perdiz, coz y luz.

Como sabes, las palabras terminadas en Z forman el plural cambiando la Z por CES. Si, por ejemplo, la palabra es PERDIZ, hay que formar una nueva palabra sin la Z:

```
PERDI
```

A la cadena "PERDI" hay que sumarle ahora la cadena "CES":

```
"PERDI" + "CES"
```

Si la palabra tiene una longitud L, es decir, si tiene L letras, la instrucción:

```
LET Q$ = LEFT$(P$,L-1)
```

carga en Q$ toda la palabra menos la última letra.

El plural se forma sin más que sumar a Q$ la cadena "CES".

```
   10 REM PLURALES
   20 INPUT "CUAL ES LA PALABRA";P$
   30 LET L=LEN(P$)
(1)40 LET Q$=LEFT$(P$,L-1)
   50 PRINT "SU PLURAL ES: ";Q$+"CES"
   60 END
```

Damos por supuesto que metemos palabras que acaban en Z. De lo contrario, como el programa no tiene ningún control de ellas, los plurales dan palabras sin sentido.

---

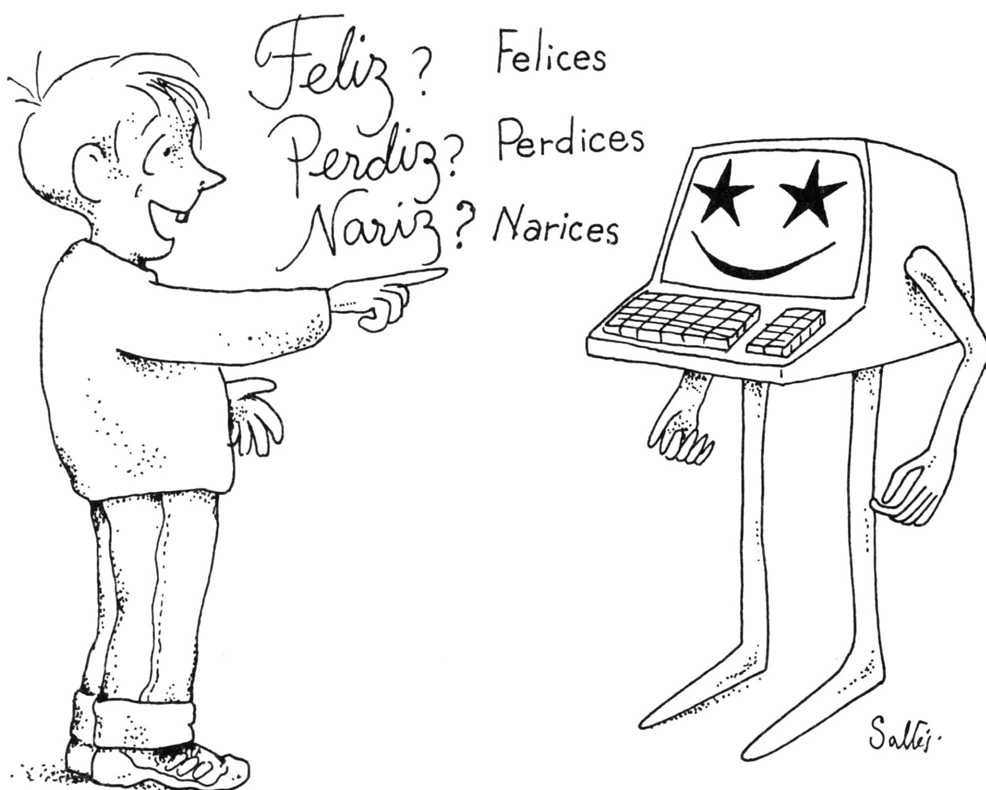
(1) Si tu ordenador es un Spectrum, sustituye la línea 40 por ésta:

```
40 LET Q$=P$(1 TO L-1)
```

Para ejercer ese control puedes intercalar en el programa anterior las líneas siguientes:

(1)
```
23 LET T$=RIGHT$(P$,1)
25 IF T$<>"Z" THEN PRINT "SE EQUIVOCO":GO
   TO 20
```

En la línea 23, con la función RIGHT$, el ordenador toma la última letra de la palabra. Si es distinta de Z, pide una nueva palabra y si es igual, continúa el programa y forma el plural.



(1) Si tu ordenador es un Spectrum, sustituye la línea 23 por ésta:

```
23 LET T$=P$(L TO L)
```

80. INICIAL DEL NOMBRE (20.4)

Diseña un programa para que el ordenador pregunte el nombre y los dos apellidos de una persona; y como respuesta escriba la inicial del nombre y los apellidos completos. (Si le das RAFAEL ALONSO PEREZ, escribirá R. ALONSO PEREZ.)

En la variable N$ cargamos el nombre y los dos apellidos. Sea por ejemplo:

```
N$ = "RAFAEL ALONSO PEREZ"
            ↑      ↑
       espacio en blanco
```

Con un ciclo FOR-NEXT recorremos la cadena N$ para que la función MID$ tome una a una las letras de N$.

```
LET A$ = MID$(N$,I,1)
```

Cuando A$ sea un espacio en blanco (A$ = " "), a su derecha están los dos apellidos (en el supuesto de que el nombre no sea compuesto). La función RIGHT$ coge todas las letras que hay a la derecha de ese espacio en blanco.

La instrucción:

```
LET P$ = RIGHT$(N$,L-I)
```

selecciona los dos apellidos, siendo L la longitud de toda la cadena, e I la posición del primer espacio en blanco. En nuestro problema L = 19 e I = 7.

```
N$ = "RAFAEL ALONSO PEREZ"
            ↑            ↖
       posición 7    posición 19
```

```
10 REM INICIAL DEL NOMBRE
20 INPUT "NOMBRE Y APELLIDOS";N$
30 LET L=LEN(N$)
```

```
(1) 40 LET I$=LEFT$(N$,1):REM INICIAL
    50 FOR I=1 TO L
(1) 60 LET A$=MID$(N$,I,1)
    70 IF A$=" " THEN GOTO 90
    80 NEXT I
(1) 90 LET P$=RIGHT$(N$,L-I)
    100 PRINT I$+". "+P$
    110 END
```

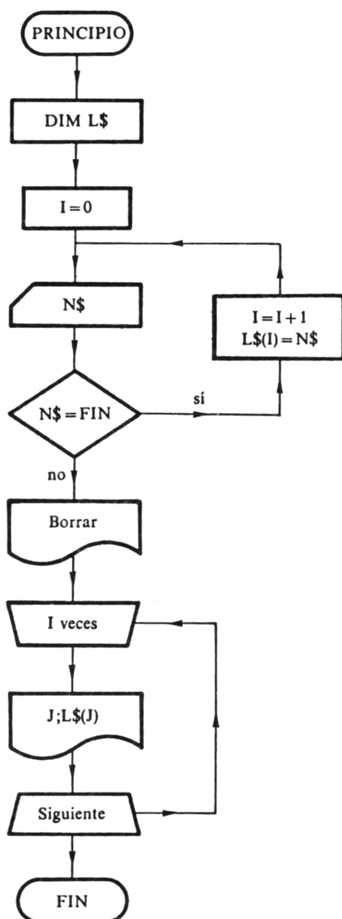
(1) Si tu ordenador es un Spectrum sustituye las líneas 40, 60 y 90 por las siguientes:

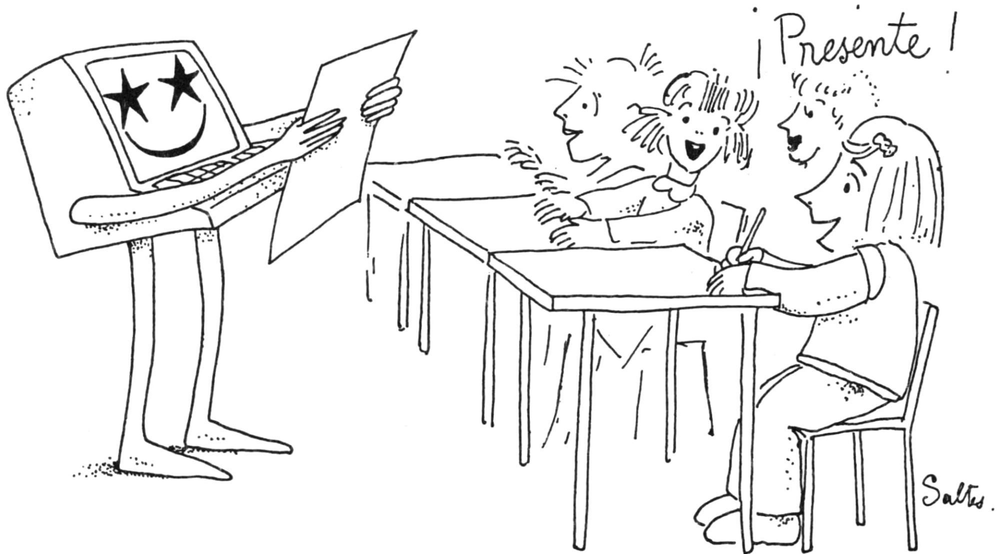
```
40 LET I$=N$(1 TO 1)
60 LET A$=N$(I TO I)
90 LET P$=N$(I+1 TO L)
```

81. LISTA DE CLASE (21.1)

Prepara un programa que cargue los nombres de tus compañeros de clase y que luego escriba la lista completa.

Como ves, el programa debe tener dos partes: en la primera el ordenador, tras preparar (dimensionar) una lista, preguntará los nombres de tus compañeros y los irá guardando en ella, hasta que le digas FIN; en la segunda, borrará la pantalla y escribirá toda la lista de una sola vez. El diagrama de flujo puede ser éste:





El programa quedará así:

```
   10 REM LISTA DE CLASE
   20 REM PREPARAR LAS VARIABLES
(1)30 DIM L$(50)
   40 LET I=0
   100 REM CREAR LA LISTA
   110 INPUT "NOMBRE";N$
   120 IF N$<>"FIN" THEN LET I=I+1:LET L$(I)
       =N$:GOTO 110
   200 REM ESCRIBIR LA LISTA
(2)210 (BORRAR LA PANTALLA)
   220 FOR J=1 TO I
   230 PRINT J;TAB(3);L$(J)
   240 NEXT J
   300 END
```

---

(1) Para Sinclair poner DIM L$ (5Ø,3Ø) para que los nombres puedan ser hasta de 3Ø letras.

(2) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT "[SHIFT] [CLR HOME]".

82. NOTAS MEDIAS (21.2)

En la lista M(I) están las notas de matemáticas de once compañeros tuyos, y en la L(J) las notas de lengua. Haz un programa que calcule la media de las notas de matemáticas y la de las notas de lengua.

En este ejercicio vamos a suponer que las notas están guardadas en líneas DATA (líneas 100 a 130). El ordenador las leerá para llenar las listas (líneas 200 a 270). Luego sumará las once notas de cada asignatura; por un lado las de matemáticas (en SM, Suma de Matemáticas) y por otro lado las de lengua (en SL, Suma de Lengua). Por último, se dividirán estas sumas por el número de compañeros para obtener las medias (líneas 300 a 340).

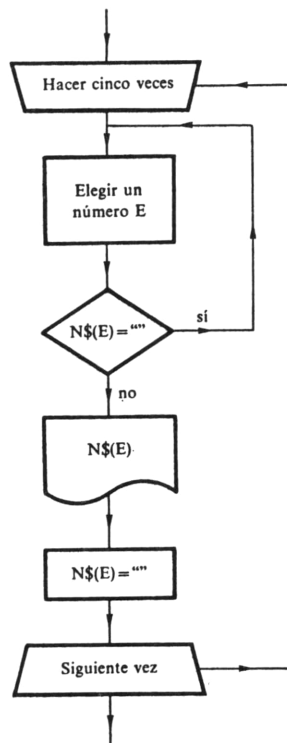
```
10 REM NOTAS MEDIAS
100 REM NOTAS DE MATEMATICAS
110 DATA 8,7,9,6,8,7,7,6,10,7,7
120 REM NOTAS DE LENGUA
130 DATA 7,6,7,6,8,9,7,9,6,6,8
200 REM LLENAR LAS LISTAS
210 DIM M(11):DIM L(11)
220 FOR C=1 TO 11
230 READ M(C)
240 NEXT C
250 FOR C=1 TO 11
260 READ L(C)
270 NEXT C
300 REM CALCULAR LAS MEDIAS
310 LET SM=0:LET SL=0
320 FOR C=1 TO 11
330 LET SM=SM+M(C):LET SL=SL+L(C)
340 NEXT C
400 REM SALIDA DE RESULTADOS
410 LET A$="  NOTA MEDIA DE "
420 PRINT:PRINT A$+"MATEMATICAS=";SM/11
430 PRINT:PRINT A$+"LENGUA=";SL/11
440 END
```

83. EQUIPO DE BALONCESTO (21.3)

Mete en una lista los nombres de doce amigos. Haz un programa para que el ordenador seleccione los nombres de cinco de ellos para formar un equipo de baloncesto.

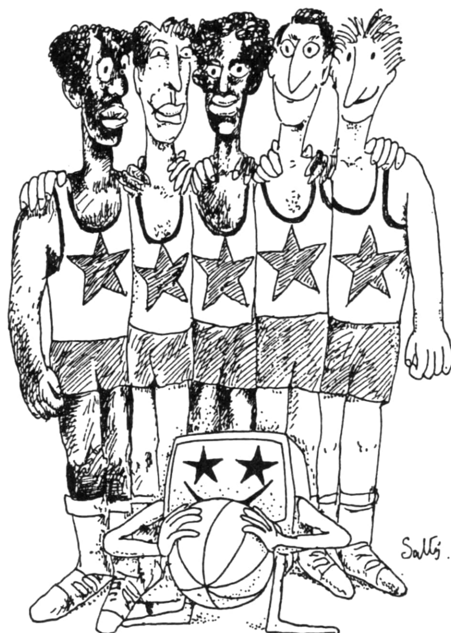
Igual que los anteriores, este programa tendrá una primera parte consistente en dimensionar una lista y llenarla con los doce nombres. Puedes dárselos con INPUT, como en el ejercicio 81, o ponerlos en líneas DATA para que el ordenador los lea todos seguidos, como en el ejercicio 82. Nosotros hemos optado aquí por el primer método.

La segunda parte consiste en escribir los nombres de los cinco jugadores del equipo. Para evitar que se repitan, cada vez que se saque un nombre se quitará de la lista de espera poniendo en su lugar la cadena vacía. Un diagrama de flujo ayudará a explicar cómo hacerlo:



El programa puede ser así:

```
   10 REM EQUIPO DE BALONCESTO
   100 REM LISTA DE NOMBRES
(1)110 DIM N$(12)
   120 FOR A=1 TO 12
   130 INPUT "NOMBRE";N$(A)
   140 NEXT A
   200 REM SELECCIONAR EL EQUIPO
   210 FOR J=1 TO 5
(2)220 LET E=1+INT(12*RND(4))
   230 IF N$(E)="" THEN GOTO 220:REM ESTE YA
       SALIO
   240 PRINT "JUGADOR ";J;" ";N$(E)
   250 LET N$(E)=""
   260 NEXT J
   300 END
```



(1) Si tu ordenador es un Spectrum, sustituye la línea 110 por ésta:

```
110 DIM N$(12,20)
```

(2) Si tu ordenador es un Spectrum, sustituye la línea 220 por estas dos:

```
20 RANDOMIZE
220 LET E=1+INT(12*RND)
```

84. ELECCIONES (21.4)

Las disposiciones electorales españolas prevén la aplicación de la ley d'Hont a la hora de distribuir los escaños entre los partidos según los votos que se hayan obtenido. El motivo de hacerlo así es evitar la dispersión del electorado en muchos partidos pequeños.

El siguiente programa, después de preguntar los datos necesarios, calcula los escaños que corresponden a cada partido en unas elecciones. Ejecútalo varias veces. Puedes mejorarlo para que sea más rápido.

```
   10 REM ELECCIONES
   20 INPUT "NUMERO DE PARTIDOS";P
   30 INPUT "ESCAÑOS A DISTRIBUIR";E
(1)40 DIM P$(P):DIM V(P):DIM A(P):DIM E(P)
   50 REM ENTRADA DE DATOS
   60 FOR I=1 TO P
   70 INPUT "NOMBRE DEL PARTIDO";P$(I)
   80 INPUT "VOTOS OBTENIDOS";V(I)
   90 NEXT I
   100 REM REPARTO DE LOS ESCAÑOS
   110 FOR X=1 TO E
   120 REM VARIABLE AUXILIAR
   130 FOR I=1 TO P
   140 LET A(I)=V(I)/(E(I)+1)
   150 NEXT I
   160 REM CALCULO DEL MAXIMO
   170 LET M=A(1):LET L=1
   180 FOR I=2 TO P
   190 IF A(I)>M THEN LET M=A(I):LET L=I
   200 NEXT I
   210 REM EL MAXIMO ESTA EN L
   220 REM SE LE AÑADE UN ESCAÑO
   230 LET E(L)=E(L)+1
   240 NEXT X
   250 REM PRESENTACION DE RESULTADOS
   260 FOR I=1 TO P
   270 PRINT P$(I),V(I);TAB(26);E(I)
   280 NEXT I
   290 END
```

(1) Si tu ordenador es un Spectrum, sustituye la línea 40 por ésta:

```
40 DIM P$(P,20):DIM V(P):DIM A(P):DIM E(P)
```

Lo mejor para entender y tratar de mejorar el programa es que lo trabajemos siguiendo sus instrucciones con ayuda de una calculadora, como si nosotros fuéramos el ordenador.

En la Comunidad Autónoma de Compulandia tienen que elegir 27 representantes para su Parlamento. El escrutinio de las elecciones arroja el siguiente resultado:

Partido	Número de votos
1. P. Junior	547.629
2. P. Reformista Reformado	402.191
3. Partido de Fútbol	342.607
4. P. Demócrata Radical	109.455
5. Partido-por-dos	87.832

Suponemos que ya hemos introducido los datos en el programa, y nos situamos en la línea 100. Para repartir cada escaño el ordenador trabaja con la variable auxiliar A(I), que resulta de dividir los votos obtenidos entre los escaños ya repartidos más uno (bucle de las líneas 130 a 150). Por ejemplo en Compulandia, cuando ya han repartido diez escaños, la situación está así:

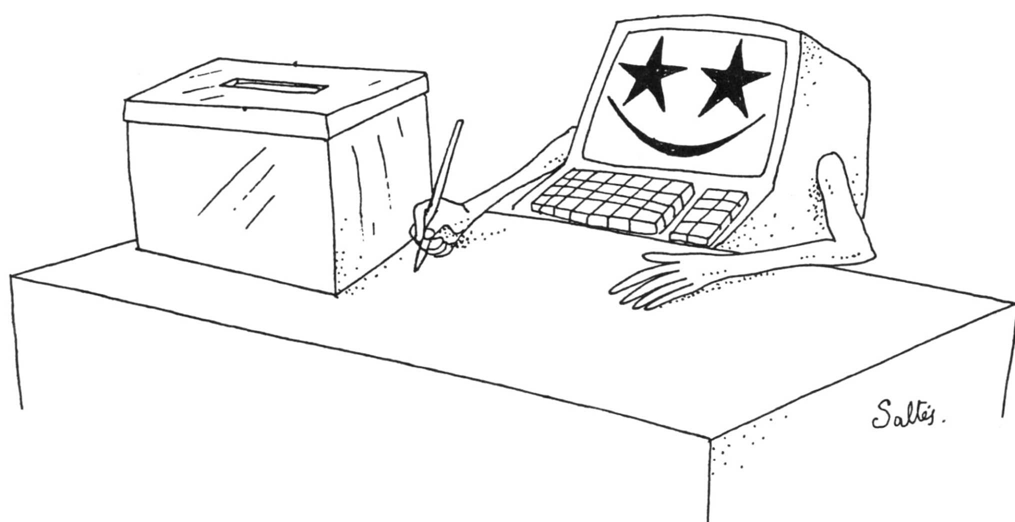
Partido	N.º votos	Variable auxiliar	Escaños
1. P. Junior	547.629	109.525,8	4
2. P. Reformista Reformado	402.191	100.547,75	3
3. Partido de Fútbol	342.607	85.651,75	3
4. P. Demócrata Radical	109.455	109.455	0
5. Partido-por-dos	87.832	87.832	0

por lo que el escaño número once se le da al partido JUNIOR, que es en este instante el que tiene más alto el valor de la variable auxiliar.

Tras la adjudicación del escaño número once, el quinto del Partido JUNIOR, se modifica su variable auxiliar, que pasa a valer:

$$\frac{547.629}{5 + 1} = 91.271,5$$

Como ves, tras la asignación de cada escaño, sólo hay que modificar la variable auxiliar de un partido. El programa que propone el enunciado re-



calcula cada vez todas las variables auxiliares (bucle de las líneas 130 a 150) y eso es un gasto de tiempo. Podemos sacar ese bucle fuera del reparto de escaños, para que sólo nos dé el valor inicial de la variable auxiliar. Luego, durante el reparto, bastará añadir la línea:

```
235 LET A(L)=V(L)/(E(L)+1)
```

La parte modificada del programa queda así:

```
100 REM VARIABLE AUXILIAR
110 FOR I=1 TO P
120 LET A(I)=V(I)
130 NEXT I
140 REM REPARTO DE LOS ESCAÑOS
150 FOR X=1 TO E
160 REM CALCULO DEL MAXIMO
170 LET M=A(1):LET L=1
180 FOR I=2 TO P
190 IF A(I)>M THEN LET M=A(I):LET L=I
200 NEXT I
210 REM EL MAXIMO ESTA EN L
220 REM SE LE ANADE UN ESCAÑO
230 LET E(L)=E(L)+1
235 LET A(L)=V(L)/(E(L)+1)
240 NEXT X
```

85. CUADRO NUMERICO (22.1)

El cuadro numérico:

```
1  1  1  1  1  1  1  0  1
1  0  0  0  0  0  0  0  1
1  0  1  1  1  0  1  1  1
1  0  0  0  1  0  1  0  1
1  1  1  0  1  0  1  0  1
1  0  0  0  1  0  0  0  1
1  0  1  1  1  1  1  1  1
```

se carga en la tabla T(I,J) con el siguiente programa:

```
10 REM CUADRO NUMERICO
20 DIM T(7,9)
30 FOR I=1 TO 7
40 FOR J=1 TO 9
50 READ T(I,J)
60 NEXT J
70 NEXT I
80 DATA 1,1,1,1,1,1,1,0,1
90 DATA 1,0,0,0,0,0,0,0,1
100 DATA 1,0,1,1,1,0,1,1,1
110 DATA 1,0,0,0,1,0,1,0,1
120 DATA 1,1,1,0,1,0,1,0,1
130 DATA 1,0,0,0,1,0,0,0,1
140 DATA 1,0,1,1,1,1,1,1,1
```

Añade otro ciclo anidado parecido al anterior que imprima en la pantalla el cuadro numérico. (Entre las líneas NEXT J y NEXT I necesitarás poner una instrucción PRINT.)

El ciclo de las líneas 30 a 70 hace siete veces lo mismo: leer los nueve números que hay en cada línea del cuadro (líneas 40, 50 y 60). Para escribirlo ordenadamente en la pantalla, hay que proceder de un modo similar: escribir en una misma horizontal los nueve números de cada línea, y eso repetirlo siete veces. Al programa de arriba hay que añadirle estas líneas:

```
200 REM ESCRIBIR EL CUADRO
230 FOR I=1 TO 7
240 FOR J=1 TO 9
250 PRINT T(I,J);
260 NEXT J
265 PRINT
270 NEXT I
300 END
```

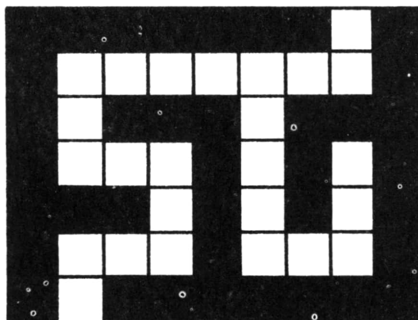
Al final de la línea 250 hay un punto y coma (;) para recordar al ordenador que debe continuar escribiendo en la misma línea; así salen los nueve números seguidos. Para que los siguientes nueve números salgan en la línea de abajo, tienes que poner la instrucción:

```
265 PRINT
```

Prueba a quitar esta orden o el punto y coma de la línea 250 y mira qué pasa.

86. LABERINTO (22.2)

El cuadrado numérico del ejercicio anterior es el código secreto de este laberinto:



en el que el 1 significa "pared" y el 0 "paso libre". Modifica el programa del ejercicio anterior para que, en lugar de escribir el cuadrado numérico, el ordenador dibuje el laberinto.

Cambia algún número en las líneas DATA y observa el efecto producido en el dibujo.

Como te indica el enunciado, para dibujar el laberinto en lugar del cuadro numérico, basta con un pequeño cambio: no hay que escribir el número de cada lugar, sino pintar un cuadrado, si el número es uno, y dejar el sitio sin pintar, si el número es cero. Eso se puede hacer sustituyendo la línea 250 por éstas:

```
250 IF T(I,J)=1 THEN PRINT "■";
255 IF T(I,J)=0 THEN PRINT " ";
```

Si tu ordenador es un Spectrum, puedes optar por esta versión:

```
250 IF T(I,J)=1 THEN PAPER=0
253 IF T(I,J)=0 THEN PAPER=7
256 PRINT " ";
```

En las líneas 250 y 253 se elige el color con que se va a pintar y en la 256 se pinta un cuadrado con el color elegido.

Si posees un Commodore, entonces debes poner lo siguiente:

```
250 IF T(I,J)=1 THEN PRINT ".[CTRL] 9  [CTRL]
    0 ";
255 IF T(I,J)=0 THEN PRINT "  ";
```

Apretando las teclas [CTRL] 9 se prepara la opción REVERSE, que pintará en contraste invertido el cuadradito siguiente, y con [CTRL] 0 se desconecta esa opción, con lo cual vuelve a pintar normal.

87. HOTEL (22.3)

En un hotel de cinco plantas las habitaciones tienen un número de dos cifras, la primera corresponde a la planta y la segunda a su situación en el pasillo. Un corte esquemático del edificio es éste:

```
51  52  53  54  55  56  57
41  42  43  44  45  46  47
31  32  33  34  35  36  37
21  22  23  24  25  26  27
11  12  13  14  15  16  17
    R E C E P C I O N
```

Haz un programa que lo represente en la pantalla del ordenador.

Este ejercicio es muy parecido al 85. Allí construíamos una tabla leyendo números almacenados en líneas DATA, y ahora hay que poner en la tabla valores que dependen del número de la habitación y del piso en que está; para la habitación 3 del piso segundo (habitación 23), por ejemplo, el valor es 2*10+3, y lo mismo para las restantes.

Después de llenar la tabla hay que enseñarla, y eso ya hemos visto cómo se hace (ejercicio 85). La única diferencia con los casos anteriores está en que, esta vez, vamos a comenzar con la última fila (la de la quinta planta) y a terminar con la primera (la del primer piso), como señala la línea 230 de este programa:

```
10 REM HOTEL
20 DIM T(5,7):REM CINCO PISOS Y SIETE HAB
   ITACIONES
25 REM LLENAR LA TABLA
30 FOR P=1 TO 5:REM LOS CINCO PISOS
40 FOR H=1 TO 7:REM LAS SIETE HABITACIONE
   S
50 LET T(P,H)=10*P+H
60 NEXT H
70 NEXT P
200 REM ESCRIBIR EL CUADRO
210 REM SE EMPIEZA POR LOS PISOS ALTOS
230 FOR P=5 TO 1 STEP -1
```

```
240 FOR H=1 TO 7
250 PRINT T(P,H);
260 NEXT H
265 PRINT
270 NEXT P
280 REM PLANTA BAJA
290 PRINT TAB(6);"R E C E P C I O N"
300 END
```

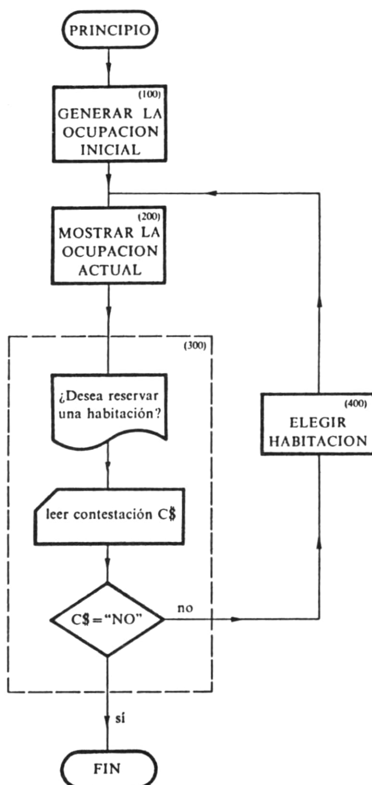
Si te parece que los números de las habitaciones salen demasiado pegados, puedes sustituir la línea 250 por esta otra:

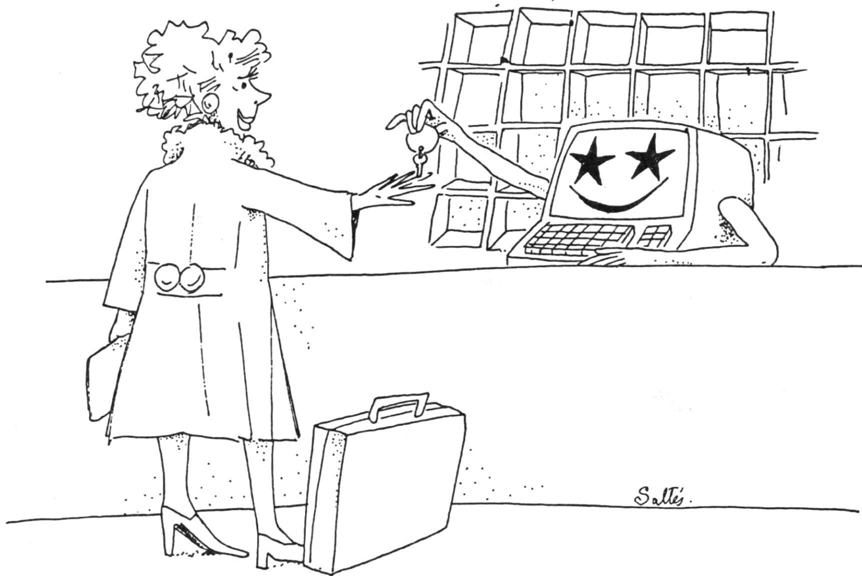
```
250 PRINT TAB(3*H);T(P,H);
```

88. RESERVA DE HABITACION (22.4)

El ordenador de la recepción del hotel emplea una tabla para saber cuáles de las habitaciones están libres y cuáles ocupadas. Prepara un programa que ponga un cero *a las habitaciones libres y un* uno *a las ocupadas, permitiendo reservar así plazas o dejar habitaciones libres, sin más que indicar el piso y la puerta.*

Vamos a emplear una tabla R de cinco filas y siete columnas, en la que inicialmente reservaremos habitaciones distribuyendo ceros y unos al azar, con el mismo significado que en el programa 86 del laberinto. Tras crear esa tabla, el ordenador va a preguntar si se desea reservar alguna habitación, y en caso afirmativo permitirá hacerlo, repitiendo el proceso hasta que ya no se quieran hacer más reservas. Un diagrama de flujo que refleja este proceso es el siguiente:






```
   10 REM RESERVA DE HABITACIONES
   20 REM
   30 REM
   100 REM GENERAR LA OCUPACION INICIAL
   110 DIM R(5,7)
   120 FOR P=1 TO 5:REM LOS CINCO PISOS
   130 FOR H=1 TO 7:REM LAS SIETE HABITACION
       ES
(1)140 LET AZ=RND(1):IF AZ<.5 THEN LET R(P,H
       )=1
   150 IF AZ>.5 THEN LET R(P,H)=0
   160 NEXT H:REM LA HABITACION SIGUIENTE
   170 NEXT P:REM EL PISO SIGUIENTE
   180 REM
   190 REM
   200 REM MOSTRAR LA OCUPACION ACTUAL
(2)210 (BORRAR LA PANTALLA)
```

---

(1) Si tu ordenador es un Spectrum, sustituye la línea 140 por estas dos:

```
115 RANDOMIZE
140 LET AZ=RND:IF AZ<.5 THEN LET R(P,H)=1
```

(2) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT "[SHIFT] [CLR HOME]".

```
220 FOR P=5 TO 1 STEP -1
230 FOR H=1 TO 7
240 PRINT TAB(3*H);R(P,H);
250 NEXT H
260 PRINT
270 NEXT P
280 PRINT TAB(6);"R E C E P C I O N"
290 REM
295 REM
300 REM SE QUIERE EFECTUAR RESERVA?
310 INPUT "DESEA RESERVAR UNA HABITACION"
    ;C$
320 IF C$="NO" THEN GOTO 510:REM AL FINAL
330 REM EL PROGRAMA CONTINUA PARA EFECTUA
    R LA RESERVA
340 REM
350 REM
400 REM ELEGIR HABITACION
410 PRINT:PRINT:PRINT
420 PRINT "ESCRIBA EL PISO Y NUMERO DE HA
    BITACION QUE DESEA (SEPARADOS POR COM
    A)";
430 INPUT P,H
440 REM SI ESTA LIBRE, RESERVARLA
450 IF R(P,H)=0 THEN LET R(P,H)=1:GOTO 48
    0
460 REM SI VIENE POR AQUI, ESTA OCUPADA
470 PRINT "LO SIENTO. ESTA OCUPADA.":GOTO
    490
480 PRINT "RESERVA EFECTUADA."
490 FOR I=1 TO 2000:NEXT I:REM_ESPERA
500 GOTO 200:REM VOLVER A ENSEÑAR LAS RES
    ERVAS
510 END
```

89. LOPE DE VEGA (23.1)

Hay una cancioncilla de Lope de Vega que dice así:

La Virgen de la Cabeza,
¡Quién como ella!
hizo gloria aquesta tierra.
¡Quién como ella!
Tiene la frente de perlas
¡Quién como ella!
y de oro fino las hebras.
¡Quién como ella!

Utiliza una subrutina para programar la escritura de estos versos.

Como ves hay cuatro versos. Puedes emplear una subrutina que escriba ese verso repetido, con lo cual el programa quedará así:

```
20 REM LOPE DE VEGA
40 PRINT "LA VIRGEN DE LA CABEZA,":GOSUB
   300
60 PRINT "HIZO GLORIA AQUESTA TIERRA.":GO
   SUB 300
80 PRINT "TIENE LA FRENTE DE PERLAS":GOSU
   B 300
100 PRINT "Y DE ORO FINO LAS HEBRAS.":GOS
    UB 300
120 END
300 REM VERSO REPETIDO
310 PRINT "!QUIEN COMO ELLA!"
320 RETURN
```

Observa que si no pones la línea 120, el verso repetido aparece cinco veces, en lugar de cuatro, y además sale un mensaje de error, porque entras en la subrutina sin querer y sin emplear la instrucción GOSUB.

90. NUMERAR LAS LINEAS (23.2)

El programa que sigue tiene por objeto tomar dos números y escribir la suma de ellos, la suma de sus cuadrados y la suma de sus cubos. Nosotros damos las líneas del programa; tú tienes que numerarlas, poner la dirección en las instrucciones GOSUB y, por último, probar el programa.

```
REM SUMAS
REM PROGRAMA PRINCIPAL
INPUT "DAME DOS NUMEROS";X,Y
LET A=X
LET B=Y
GOSUB
LET A=X*X
LET B=Y*Y
GOSUB
LET A=X*X*X
LET B=Y*Y*Y
GOSUB
END
```

```
REM SUBRUTINA PARA SUMAR
LET S=A+B
PRINT "LA SUMA ES";S
RETURN
```

Esta es una posibilidad:

```
5 REM SUMAS
10 REM PROGRAMA PRINCIPAL
15 INPUT "DAME DOS NUMEROS";X,Y
20 LET A=X
25 LET B=Y
30 GOSUB 100
35 LET A=X*X
40 LET B=Y*Y
```

```
45 GOSUB 100
50 LET A=X*X*X
55 LET B=Y*Y*Y
60 GOSUB 100
65 END
100 REM SUBRUTINA PARA SUMAR
105 LET S=A+B
110 PRINT "LA SUMA ES";S
115 RETURN
```

91. JUEGO DE DADOS (23.3)

La programación modular es una técnica que consiste en descomponer un problema en varias partes llamadas módulos, *programar por separado cada una de ellas, y por último, unirlas. Cada módulo se puede programar como una subrutina que es reclamada desde el programa principal, cuya misión es unir los distintos módulos.*

En la programación de un juego puedes distinguir tres partes:

1.ª PRESENTACION DEL JUEGO, que tiene por objeto explicar al usuario las reglas del juego.

2.ª JUEGO propiamente dicho.

3.ª FINAL DEL JUEGO, en donde el ordenador pregunta si el jugador quiere volver a jugar, en cuyo caso comienza otra vez el juego, o si quiere terminar.

El juego que tienes que programar con esta técnica es el siguiente:

a) Tirar dos dados.
b) Si la suma es 7 u 11, gana el ordenador; si no, ganas tú.

Te damos el programa principal. Tú tienes que escribir las tres subrutinas:

```
10 REM DADOS
20 REM PROGRAMA PRINCIPAL
30 GOSUB 100
40 GOSUB 200
50 GOSUB 300
60 IF R$="S" THEN GOTO 40
70 END
```

```
100 REM SUBRUTINA DE PRESENTACION
..................................................................
..................................................................
199 RETURN
```

```
200 REM JUEGO
.............................................................
.............................................................
299 RETURN
```

```
300 REM FINAL DE LA JUGADA
.............................................................
.............................................................
399 RETURN
```

Ya habrás observado que en el enunciado el juego está descompuesto en varios módulos. Cada módulo es una subrutina. La primera subrutina quedará así:

```
   100 REM SUBRUTINA DE PRESENTACION
   110 REM EXPLICACION
   120 PRINT:PRINT
   130 PRINT "HOLA.":PRINT
   140 PRINT "VAMOS A JUGAR A LOS DADOS, TIR
       ARE DOS DADOS..."
   150 PRINT "...SI LOS RESULTADOS SUMAN 7 U
       11, GANO YO..."
   160 PRINT "...SI LA SUMA ES OTRA, GANAS T
       U, AMO."
   170 PRINT "PARA CONTINUAR PULSA UNA TECLA"
(1)180 GET A$:IF A$="" THEN GOTO 180
   199 RETURN
```

La subrutina de juego es la siguiente:

```
   200 REM JUEGO
(2)205 (BORRAR LA PANTALLA)
```

(1) Si tu ordenador es un Spectrum, sustituye la línea 180 por ésta:

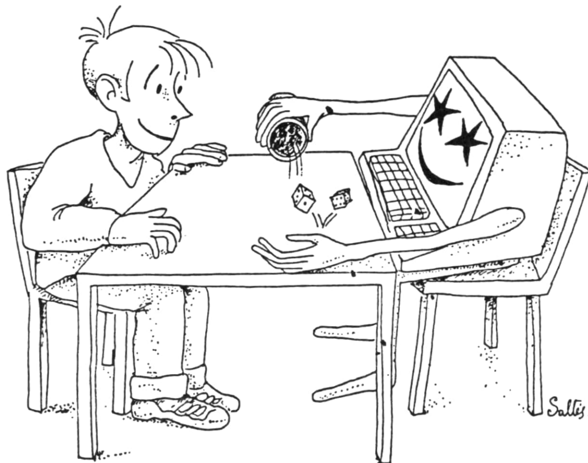
```
180 LET A$=INKEY$:IF A$="" THEN GOTO 180
```

(2) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT "(SHIFT) (CLR HOME)".

```
   215 LET SU=0:REM LA SUMA AUN NO VALE NADA
   220 FOR I=1 TO 2:REM DOS VECES
(1)225 LET D=1+INT(RND(3)*6):REM SE TIRA EL
       DADO
   230 PRINT:PRINT "CON EL DADO";I;"HA SALID
       O";D
   235 LET SU=SU+D
   240 NEXT I
   245 PRINT:PRINT "LA SUMA ES";SU
   250 IF SU=7 OR SU=11 THEN PRINT "DISCULPA
       HUMANO, PERO HE GANADO YO.":GOTO 299
   260 PRINT "UNA VEZ MAS ME HAS GANADO, !OH
       SUBLIME HUMANO!"
   299 RETURN
```

Por último, falta preguntar si se va a continuar jugando. Guardamos la contestación en R$.

```
300 REM FINAL DE LA JUGADA
305 PRINT:PRINT:PRINT
310 INPUT "QUIERES SEGUIR JUGANDO (S/N)";
    R$
399 RETURN
```



(1) Si tu ordenador es un Spectrum, sustituye la línea 225 por estas dos:

```
15 RANDOMIZE
225 LET D=1+INT(RND*6)
```

92. ADIVINA (23.4)

Aplica las técnicas de programación modular para mejorar este programa:

```
   10 REM ADIVINA
(1)20 LET X=1+INT(100*RND(4))
   30 PRINT "DAME UN NUMERO";
   40 INPUT N
   50 IF N>X THEN PRINT "TE HAS PASADO":GOTO
      30
   60 IF N<X THEN PRINT "NO HAS LLEGADO":GOT
      O 30
   70 PRINT "ENHORABUENA: HAS ACERTADO"
   80 END
```

El programa realiza las siguientes tareas:

Elige un número (X)
Repite lo siguiente
  Pide que lo adivines (N)
  Indica qué tal ha estado el intento
hasta que lo aciertas
Fin de lo que Repite
FIN

No es posible acortarlo; lo que sí se puede hacer es programarlo de manera que quede más claro qué se hace en cada momento. Para ello vamos a descomponer el programa en tres módulos, que programaremos como subrutinas:

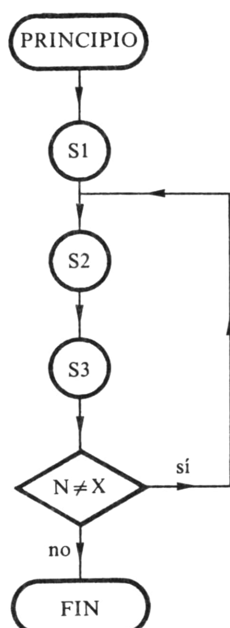
Subrutina 1: elige el número X.
Subrutina 2: te pide que lo adivines.
Subrutina 3: dice cómo ha estado el intento.

Estas subrutinas se unen en la forma que indica este diagrama de flujo:

---

(1) Si tu ordenador es un Spectrum, sustituye la línea 20 por estas dos:

```
15 RANDOMIZE
20 LET X=1+INT(100*RND)
```



El programa puede ser éste:

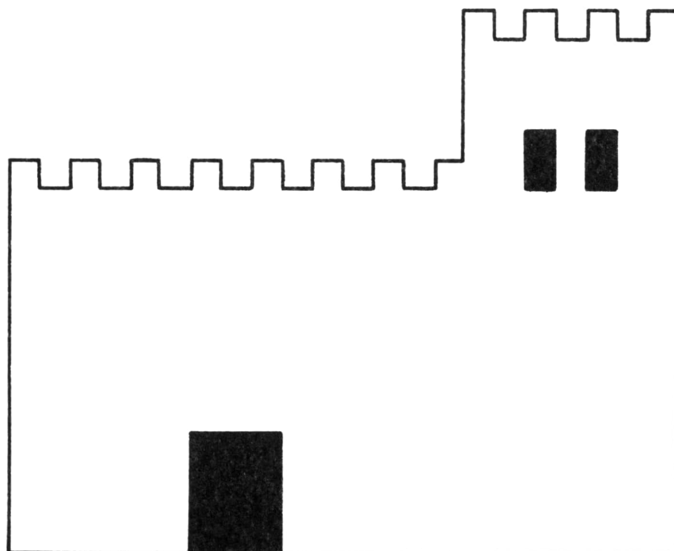
```
  10 REM ADIVINA
  20 GOSUB 1000:REM ELIGE X
  30 GOSUB 2000:REM ADIVINALO
  40 GOSUB 3000:REM QUE TAL?
  50 IF N<>X THEN GOTO 30:REM SIGUE INTENTA
     NDOLO
  60 END
  1000 REM ELIGE X
(1)1010 LET X=1+INT(100*RND(4))
  1020 RETURN
  2000 REM ADIVINALO
  2010 PRINT "DAME UN NUMERO";
  2020 INPUT N
  2030 RETURN
  3000 REM ORIENTACION
  3010 IF N>X THEN PRINT "TE HAS PASADO":GO
       TO 3040
  3020 IF N<X THEN PRINT "NO HAS LLEGADO":G
       OTO 3040
  3030 PRINT "ENHORABUENA. HAS ACERTADO"
  3040 RETURN
```

(1) Si tu ordenador es un Spectrum, sustituye la línea 1010 por estas dos:

```
15 RANDOMIZE
1010 LET X=1+INT(100*RND)
```

93. **CASTILLO** (24.1)

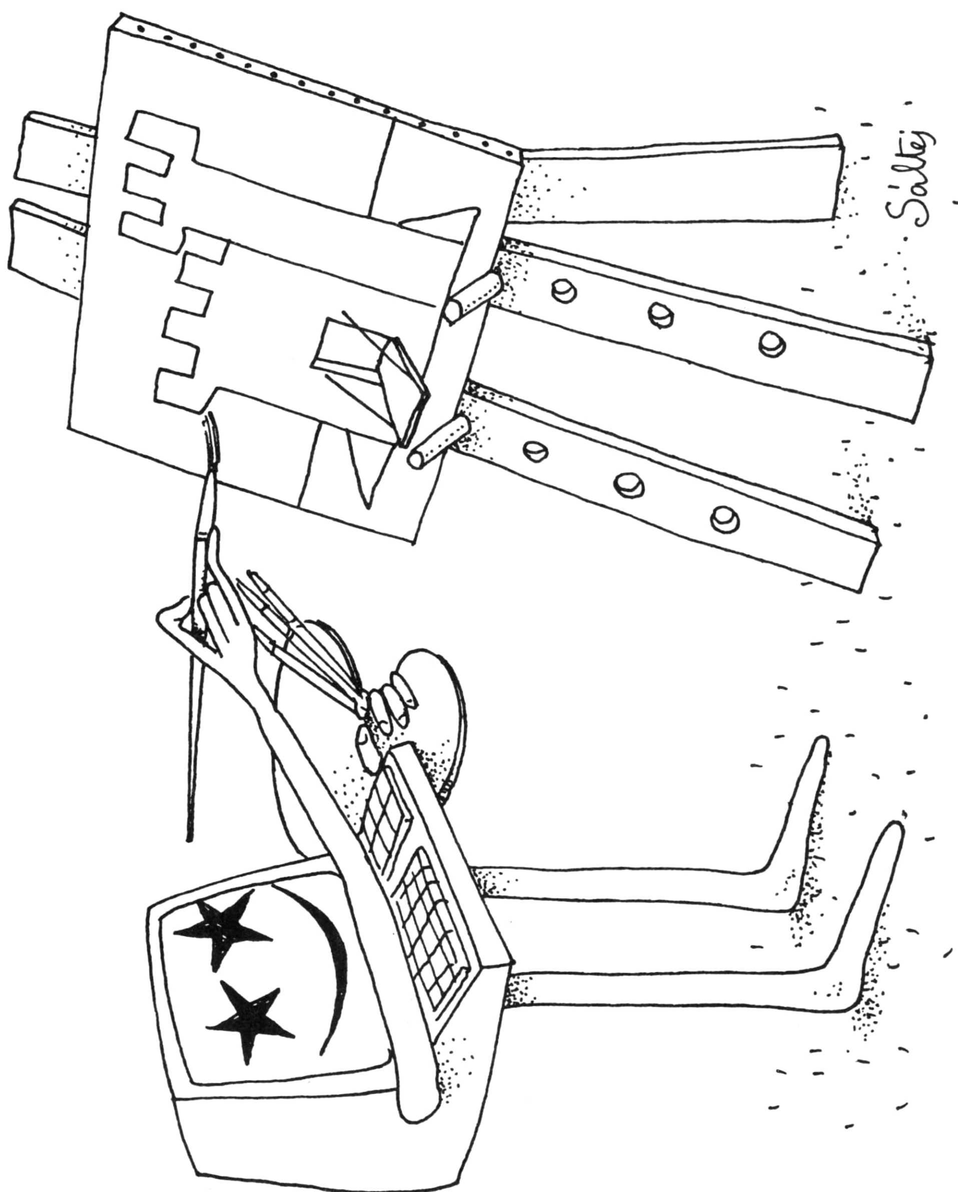
Haz un programa que pinte esta fortaleza. El suelo ha de ser verde, el cielo azul y las ventanas y puertas oscuras.



Con excepción de las almenas, todo el castillo se puede descomponer en rectángulos de diferentes colores. Para pintarlo vamos a dar los siguientes pasos:

1.º Poner todo azul claro (cielo).
2.º Poner el suelo verde, con un rectángulo.
3.º Hacer un rectángulo amarillo para el muro.
4.º Pintar otro rectángulo amarillo para la torre.
5.º Dos rectángulos negros para las ventanas.
6.º Otro para la puerta.
7.º Poner almenas sobre el muro.
8.º Poner almenas en lo alto de la torre.

Para pintar cada rectángulo fijaremos el color (K), su ancho y alto (AN y AL) y la fila y columna (FI y CO) en que debe pintarse su vértice superior



izquierdo. Te recordamos que la fila más alta es la cero y la columna más a la izquierda también la cero.

Como los diferentes ordenadores trabajan con el color de distintas maneras, vamos a resolver el problema de dos formas: primero lo haremos para ordenadores con la instrucción PRINT AT (tipo Spectrum), y luego para ordenadores que no la tienen (tipo Commodore).

Para ordenadores con la instrucción PRINT AT, la subrutina que pinta rectángulos puede ser así:

```
300 REM PINTAR RECTANGULO CON "PRINT AT"
310 PAPER K
320 REM PARA CADA I SE PINTA UNA FILA
330 FOR I=FI TO FI+AL-1
340 REM CON CADA J SE PINTA UN CUADRADO D
    E LA FILA
350 FOR J=CO TO CO+AN-1
360 PRINT AT I,J;" "
370 NEXT J
380 NEXT I
390 RETURN
```

Las almenas se pueden pintar con una subrutina parecida. También hace falta conocer el largo (AN) y primer lugar (FI y CO) de la almena, pero no su alto. La diferencia principal entre los rectángulos y las almenas está en que para los rectángulos había que pintar siempre, pero para las almenas sólo se pintan los lugares impares. La subrutina puede ser así:

```
500 REM PINTAR ALMENA CON "PRINT AT"
510 PAPER 6
540 REM CON CADA J SE PINTA UN CUADRADO D
    E ALMENA
550 FOR J=1 TO AN
560 REM SI EL LUGAR ES IMPAR (J/2)<>INT(J
    /2)
570 IF J/2<>INT(J/2) THEN PRINT AT FI,CO+
    J-1;" "
580 NEXT J
590 RETURN
```

Para el Spectrum, el programa será el siguiente:

```
10 REM CASTILLO
20 REM BORRAR LA PANTALLA
```

```
30 PAPER 5:CLS
40 FOR R=1 TO 6
50 REM LEER LOS DATOS DEL RECTANGULO
60 READ K,AN,AL,FI,CO
70 GOSUB 300:REM PINTAR RECTANGULO
80 NEXT R
100 REM AHORA LAS ALMENAS
110 FOR A=1 TO 2
120 READ AN,FI,CO
140 GOSUB 500:REM PINTAR ALMENA
150 NEXT A
160 STOP
```

...

aquí debes poner la subrutina 300

...

aquí debes poner la subrutina 500

. . .

```
600 REM DATOS CASTILLO CON "PRINT AT"
610 DATA 4,32,3,19,0:REM SUELO
620 DATA 6,15,11,9,5:REM MURO
630 DATA 6,7,15,5,20:REM TORRE
640 DATA 0,1,2,7,22:REM VENTANA IZDA.
650 DATA 0,1,2,7,24:REM VENTANA DCHA.
660 DATA 0,3,4,16,11:REM PUERTAS
700 REM DATOS ALMENAS
710 DATA 15,8,5:REM ALMENAS MURO
720 DATA 7,4,20:REM ALMENAS TORRE
```

Si tu ordenador *no* tiene la instrucción PRINT AT, por ejemplo Commodore, no necesitas pintar el cielo. En estos aparatos tienes que usar PRINT sin nada detrás para bajar a la fila en que empieza el rectángulo o almena que vayas a pintar (para bajar usaremos la subrutina 200), y luego emplearemos subrutinas parecidas a las anteriores para los dibujos.

```
200 REM ... SUB 200 ... BAJAR A LA PRIMER
    A FILA DEL DIBUJO
210 PRINT "[CLR HOME]";
220 REM SI LA FILA ES LA PRIMERA DE PANTA
    LLA, NO HAY QUE BAJAR
230 IF FI=0 THEN GOTO 280
240 REM SI LA FILA NO ES LA CERO, HAY QUE
    BAJAR
```

```
250 FOR F=1 TO FI
260 PRINT
270 NEXT F
280 RETURN
```

La subrutina que emplearemos para pintar los rectángulos, es ésta:

```
400 REM ... SUB 400 ... PINTAR RECTANGULO
410 REM PARA CADA I SE PINTA UNA FILA
420 FOR I=FI TO FI+AL-1
430 REM CON CADA J SE PINTA UN CUADRADO D
    E LA FILA
440 FOR J=CO TO CO+AN-1
450 PRINT TAB(J);K$;
460 NEXT J
470 PRINT:REM BAJAR A OTRA FILA
480 NEXT I
490 RETURN
```

Con K$ se pinta un cuadradito del color deseado. Para pintar la almena usaremos:

```
700 REM ... SUB 700 ... PINTAR ALMENA
740 REM CON CADA J SE PINTA UN CUADRADO D
    E LA ALMENA
750 FOR J=1 TO AN
760 REM SI EL LUGAR ES IMPAR, J/2<>INT(J/
    2)
770 IF J/2<>INT(J/2) THEN PRINT TAB(CO+J-
    1);K$;
780 NEXT J
790 RETURN
```

El programa total quedará así:

```
10 REM CASTILLO BIS
20 REM BORRAR LA PANTALLA
30 PRINT "[SHIFT] [CLR HOME]"
40 FOR R=1 TO 6
50 REM LEER LOS DATOS DEL RECTANGULO
60 READ K$,AN,AL,FI,CO
70 GOSUB 200:REM BAJAR
80 GOSUB 400:REM PINTAR RECTANGULO
90 NEXT R
```

```
100 REM AHORA LAS ALMENAS
110 FOR A=1 TO 2
120 READ K$,AN,FI,CO
130 GOSUB 200:REM BAJAR
140 GOSUB 700:REM PINTAR ALMENA
150 NEXT A
160 GOTO 160:REM PARA QUE NO SE ESTROPEE
    EL DIBUJO
170 END
```

. . .

aquí debes poner la subrutina 200

. . .

aquí debes poner la subrutina 400

. . .

aquí debes poner la subrutina 700

. . .

```
800 REM DATOS CASTILLO
810 DATA "[CTRL]6[CTRL]9 [CTRL]0",39,5,19,0:
    REM SUELO
820 DATA "[CTRL]8[CTRL]9 [CTRL]0",15,11,9,5:
    REM MURO
830 DATA "[CTRL]8[CTRL]9 [CTRL]0",7,15,5,20:
    REM TORRE
840 DATA "[CTRL]1[CTRL]9 [CTRL]0",1,2,7,22:R
    EM VENTANA IZDA.
850 DATA "[CTRL]1[CTRL]9 [CTRL]0",1,2,7,24:R
    EM VENTANA DCHA.
860 DATA "[CTRL]1[CTRL]9 [CTRL]0",3,4,16,11:
    REM PUERTA
900 REM DATOS ALMENA
910 DATA "[CTRL]8[CTRL]9 [CTRL]0",15,8,5:REM
    ALMENAS MURO
920 DATA "[CTRL]8[CTRL]9 [CTRL]0",7,4,20:REM
    ALMENAS TORRE
```

94. PAJAROS (24.2)

Consigue que estos pájaros muevan las alas.

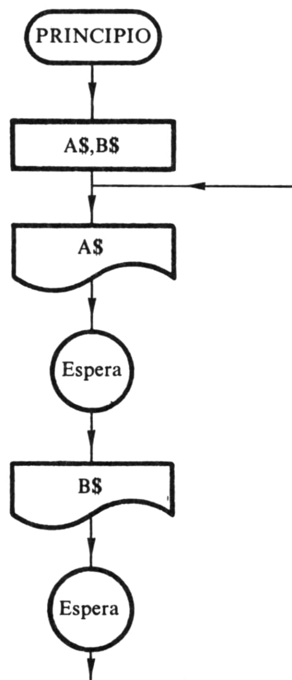




Si en tu ordenador hay teclas que dibujan directamente en la pantalla circunferencias y cuartos de circunferencia, el programa se simplifica y basta con:

Definir dos dibujos del pájaro (alas arriba y alas abajo).
Repetir indefinidamente:
  Pintar el pájaro con las alas hacia arriba.
  Esperar un poco, para que se le vea bien.
  Pintar el pájaro en el mismo sitio con las alas hacia abajo.
  Esperar un poco, para que se le vea bien.
Fin de lo que se Repite.

El diagrama es:



El programa correspondiente puede ser éste:

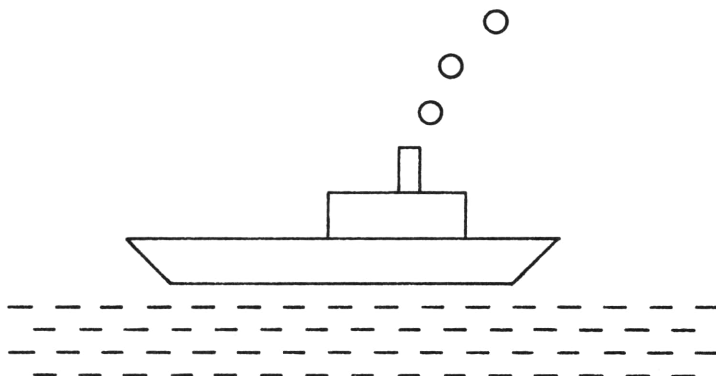
```
10 REM PAJAROS
20 PRINT "[SHIFT] [CLR HOME]":REM BORRAR LA PANTAL
   LA
30 LET A$=""
40 LET B$=""
100 REM MOVIMIENTO ALAS ARRIBA
110 PRINT " [CLR HOME] ":REM CURSOR AL ORIGEN
120 PRINT TAB(3);A$:PRINT TAB(18);A$:PRIN
    T TAB(8);A$
130 GOSUB 300:REM ESPERA
200 REM ALAS ABAJO
210 PRINT " [CLR HOME] ":REM CURSOR AL ORIGEN
220 PRINT TAB(3);B$:PRINT TAB(18);B$:PRIN
    T TAB(8);B$
230 GOSUB 300:REM ESPERA
240 GOTO 110
300 REM ESPERA
310 FOR I=1 TO 200:NEXT I
320 RETURN
```

Si tu ordenador es un Sinclair, la versión apropiada será ésta:

```
10 REM PAJAROS
20 CLS:REM BORRAR LA PANTALLA
30 LET A$=""
40 LET B$=""
100 REM MOVIMIENTO ALAS ARRIBA
110 PRINT AT 0,3;A$;AT 1,18;A$;AT 2,8;A$
120 PAUSE 10:REM ESPERA
200 REM ALAS ABAJO
210 PRINT AT 0,3;B$;AT 1,18;B$;AT 2,8;B$
220 PAUSE 10:REM ESPERA
230 GOTO 110
```

95. BARCO (24.3)

Dibuja este barquito en la pantalla. Combinando alternativamente las líneas de rayas podrás conseguir el efecto del oleaje.



Este programa constará de dos partes, el dibujo del humo y el barco, parecido a lo que ya vimos en los ejercicios 1, 2 y 3, y el oleaje en movimiento. La sensación de movimiento de las olas se puede conseguir alternando dos dibujos, como acabamos de hacer con los pájaros. Lo que necesitamos es un programa que haga lo siguiente:

Definir dos dibujos de olas.
Pintar el humo.
Pintar el barco.
Repetir indefinidamente:
    Pintar el primer dibujo de olas.
    Esperar un poco, para que se le vea bien.
    Intercambiar los dibujos del oleaje.
Fin de lo que se Repite.

Si tu ordenador tiene predefinidos los dibujos necesarios, el programa correspondiente es éste:

```
5 REM BARCO EN ALTA MAR
10 REM CREAR OLAS
```

```
15 LET A$=" - - - - - - - - - - - - - - - - - -
    _"
20 LET B$="- - - - - - - - - - - - - - - - - -
    _ "
100 REM PINTAR EL HUMO
105 PRINT "[SHIFT] [CLR HOME]":REM BORRAR LA PANTA
    LLA
110 PRINT:PRINT TAB(22);"O"
115 PRINT TAB(19);"O"
120 PRINT:PRINT TAB(17);"O"
200 REM PINTAR EL BARCO
205 PRINT TAB(3);"        ⊓"
210 PRINT TAB(3);"        ||"
215 PRINT TAB(3);"      ┌──┐"
220 PRINT TAB(3);"______|____|__"
225 PRINT TAB(3);"\            /"
230 PRINT TAB(3);" \          /"
235 PRINT TAB(3);"  __________"
300 REM OLEAJE
305 PRINT "[CLR HOME]":REM CURSOR AL ORIGEN
310 FOR I=1 TO 12:PRINT:NEXT I:REM BAJADA
315 PRINT A$:PRINT B$:PRINT A$:PRINT B$
320 FOR I=1 TO 200:NEXT I:REM ESPERA
325 LET C$=A$:LET A$=B$:LET B$=C$
330 GOTO 305
```

Si tu ordenador es un Spectrum, debes usar este otro programa:

```
5 REM BARCO EN ALTA MAR
10 REM CREAR OLAS
15 LET A$=" - - - - - - - - - - - - - - - - - -
    _"
20 LET B$="- - - - - - - - - - - - - - - - - -
    _ "
100 REM PINTAR EL HUMO
105 CLS
110 PRINT:PRINT TAB(22);"O"
115 PRINT TAB(19);"O"
120 PRINT:PRINT TAB(17);"O"
200 REM BARCO
205 PLOT 72,72:DRAW -16,16:DRAW 136,0:DRA
    W -16,-16:DRAW -104,0:REM CASCO
```

```
210 PLOT 108,88:DRAW 0,16:DRAW 48,0:DRAW
    0,-16:REM OBRA MUERTA
215 PLOT 132,104:DRAW 0,16:DRAW 8,0:DRAW
    0,-16:REM CHIMENEA
300 REM OLEAJE
305 PRINT AT 13,0;A$;AT 14,0;B$;AT 15,0;A
    $;AT 16,0;B$
310 PAUSE 10
315 LET C$=A$:LET A$=B$:LET B$=C$
320 GOTO 305
```

96. CABALLO CORREDOR (24.4)

La letra π parece un caballito. Si haces la asignación C$ = "π", y luego imprimes C$ muchas veces en la misma línea, tendrás una fila de caballos:

π π π π π π π

Pero si haces C$ = " π" (con un espacio a la izquierda de π), y luego imprimes C$ en la misma línea precedido de TAB(I); (para I = 1, 2, 3, 4...), entonces el espacio borrará el caballito anterior, y tendrás la sensación de que el caballito está corriendo. Hazlo. ¡A ver qué pasa!

Para hacer correr al caballito por una línea basta borrarlo del sitio en que estuviese pintado y situarlo un lugar más a la derecha.

```
10 REM CABALLO CORREDOR
20 FOR I=0 TO 38
30 PRINT TAB(I);" π";
40 NEXT I
50 END
```

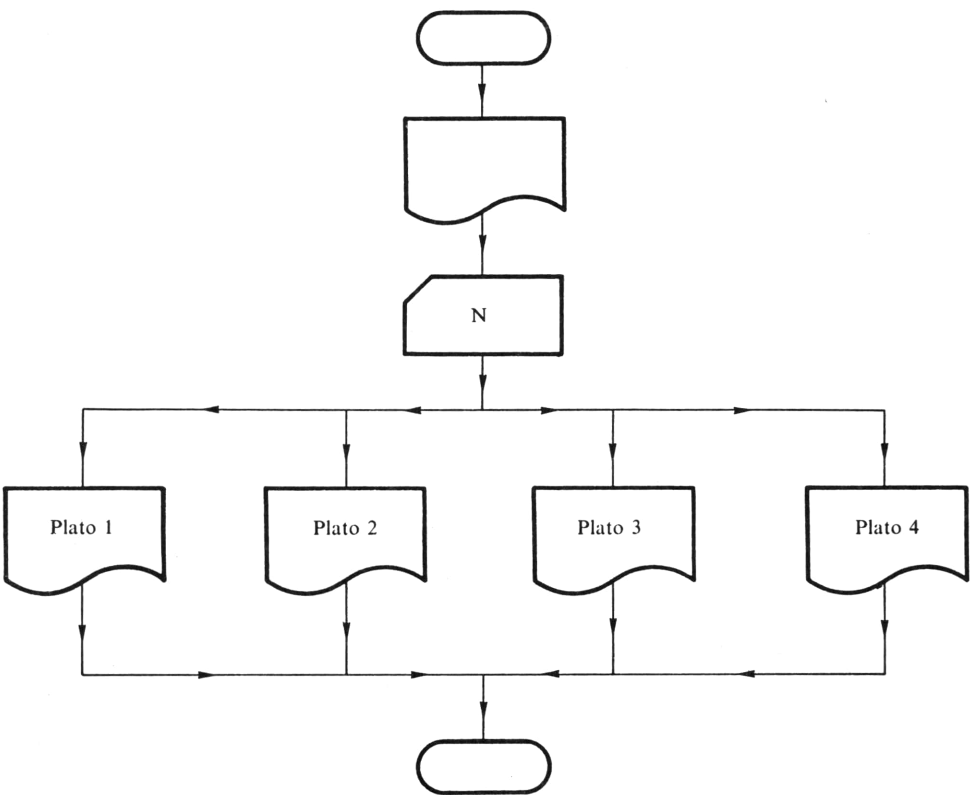
Hacemos variar I de 0 a 38 por si tu ordenador tiene 40 columnas en pantalla; si el número de columnas es menor o mayor, puedes variar este valor y ajustarlo a tu aparato.

Prueba a quitar el punto y coma (;) final de la línea 30 y observa el efecto.

97. **CAFETERIA** (25.1)

La cafetería Micro ofrece a sus clientes los platos combinados 1, 2, 3 y 4. Utiliza la instrucción ON...GOTO... para confeccionar un programa de modo que, al pulsar un número, el ordenador muestre en la pantalla la composición del plato correspondiente.

El diagrama de flujo del programa va a ser así:



Empleando la instrucción ON...GOTO... dirigiremos la ejecución del programa a las líneas que exhiben la composición del plato seleccionado. Si el número elegido es superior a cuatro, el programa acabará sin enseñar ningún plato.

```
10 REM CAFETERIA
20 REM PRESENTACION
30 GOSUB 5000:REM CABECERA
50 PRINT TAB(2);"!BIENVENIDO A SU CAFETER
   IA!"
60 PRINT:PRINT
70 PRINT "DISPONEMOS DE CUATRO FABULOSOS
   PLATOS COMBINADOS PARA QUE VD. ELIJA";
75 PRINT " EL MAS ACORDE CON SUS GUSTOS":
   PRINT
80 PRINT "SI LO DESEA PUEDE VD. SELECCION
   AR UN NUMERO, DEL 1 AL 4, PARA";
85 PRINT " CONOCER LA COMPOSICION DEL PLA
   TO CORRESPONDIENTE.":PRINT
90 INPUT "NUMERO DEL PLATO";N
100 GOSUB 5000:REM CABECERA
(1)110 ON N GOTO 1000,2000,3000,4000
120 REM SI N ES MENOR QUE 1 O MAYOR QUE 4
    , SIGUE AQUI
130 GOTO 8000:REM TERMINAR, PORQUE NO SE
    QUIERE NINGUNO DE LOS PLATOS
1000 REM PLATO 1
1010 PRINT TAB(10);"PLATO 1":PRINT
1020 PRINT TAB(5);"HUEVOS FRITOS"
1030 PRINT TAB(5);"SALCHICHAS"
1040 PRINT TAB(5);"PATATAS FRITAS"
1050 PRINT TAB(10);480;" PTS."
1060 GOTO 8000:REM TERMINAR
2000 REM PLATO 2
2010 PRINT TAB(10);"PLATO 2":PRINT
2020 PRINT TAB(5);"FILETE DE TERNERA"
2030 PRINT TAB(5);"SALCHICHAS"
2040 PRINT TAB(5);"LECHUGA"
2050 PRINT TAB(10);520;" PTS."
2060 GOTO 8000:REM TERMINAR
3000 REM PLATO 3
3010 PRINT TAB(10);"PLATO 3":PRINT
3020 PRINT TAB(5);"CALAMARES FRITOS"
3030 PRINT TAB(5);"ENSALADILLA RUSA"
3040 PRINT TAB(5);"CROQUETAS"
3050 PRINT TAB(10);475;" PTS."
3060 GOTO 8000:REM TERMINAR
4000 REM PLATO 4
```

(1) Si tu ordenador es un Spectrum, sustituye la línea 110 por ésta:

```
110 GOTO 1000*N
```

```
4010 PRINT TAB(10);"PLATO 4":PRINT
4020 PRINT TAB(5);"TORTILLA FRANCESA"
4030 PRINT TAB(5);"LECHUGA"
4040 PRINT TAB(5);"TOMATE"
4050 PRINT TAB(10);455;" PTS."
4060 GOTO 8000:REM TERMINAR
5000 REM CABECERA
(1)5010 (BORRAR LA PANTALLA)
5020 PRINT:PRINT
5030 PRINT "  CAFETERIA   MICRO"
5040 REM SUBRAYAR
5050 FOR I=2 TO 18
5060 PRINT TAB(I);"=";
5070 NEXT I
5080 PRINT:PRINT:PRINT
5090 RETURN
8000 END
```

(1) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT "SHIFT CLR HOME".

98. PASATIEMPOS (25.2)

Diseña un programa para que aparezca en la pantalla el siguiente "menú":

```
1. ADIVINANZA

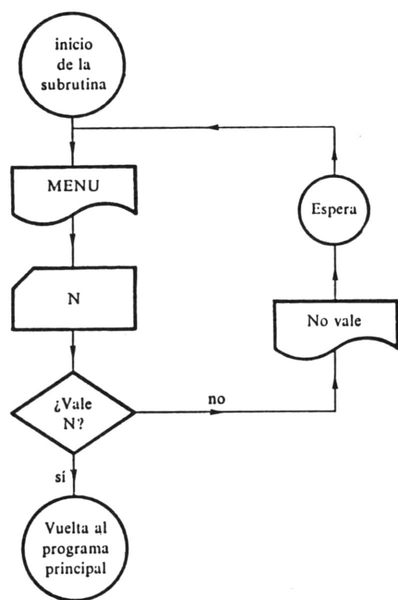
2. REFRAN

3. CHISTE

   ELIGE UN NUMERO
```

A continuación debe aparecer el texto correspondiente al número elegido.

Este ejercicio es prácticamente igual que el anterior, por lo que podemos usar el mismo diagrama de flujo; pero vamos a sustituir la primera parte del programa (Presentación y Elección) por una subrutina que contenga esas mismas partes y además compruebe si el valor elegido es válido o no, y en caso negativo le diga al "gracioso" que intentaba reírse de él (del ordenador) que tiene que repetir la Elección, porque la anterior no vale. El diagrama de flujo de esta subrutina será así:



```
  10 REM PASATIEMPOS
  20 REM PRESENTACION
  30 GOSUB 5000:REM ELECCION A PRUEBA DE LI
     STOS
(1)80 ON N GOTO 100,200,300
  100 REM ADIVINANZA
(2)110 (BORRAR LA PANTALLA)
  120 PRINT "UNA SEÑORA MUY ENSEÑORADA"
  130 PRINT "SIEMPRE VA EN UN COCHE"
  140 PRINT "Y SIEMPRE VA MOJADA"
  150 PRINT:PRINT:PRINT "?QUE ES";
  160 INPUT C$
  170 IF C$="LA LENGUA" OR C$="LENGUA" THEN
      PRINT "!MUY BIEN!":GOTO 8000:REM FIN
  180 PRINT "ESO NO ES":PRINT "PRUEBA OTRA
      VEZ.":GOSUB 6000:REM ESPERA
  190 GOTO 100
  200 REM REFRAN
(2)210 (BORRAR LA PANTALLA)
  220 PRINT "A PERRO VIEJO"
  230 PRINT "TODO SE LE VUELVEN PULGAS."
  240 GOTO 8000:REM FIN
  300 REM CHISTE
(2)310 (BORRAR LA PANTALLA)
  320 PRINT "ERA UN SEÑOR TAN BAJITO QUE CU
      ANDO SE SENTABA";
  330 PRINT " EN EL SUELO, LE COLGABAN LAS
      PIERNAS."
  340 GOTO 8000:REM FIN
  5000 REM ELECCION SEGURA
(2)5010 (BORRAR LA PANTALLA)
  5020 PRINT TAB(3);"1. ADIVINANZA"
  5030 PRINT TAB(3);"2. REFRAN"
  5040 PRINT TAB(3);"3. CHISTE"
  5050 PRINT
  5060 PRINT TAB(6);"ELIGE UN NUMERO"
  5070 INPUT N
  5080 IF N>=1 AND N<=3 THEN GOTO 5200
```

(1) Si tu ordenador es un Spectrum, sustituye la línea 80 por ésta:

```
80 GOTO 100*N
```

(2) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT "(SHIFT) (CLR HOME)".

```
5090 PRINT "ESE NUMERO NO VALE"
5100 GOSUB 6000:REM ESPERA
5110 GOTO 5000
5200 REM EL N ELEGIDO, SI VALE
5210 RETURN
6000 REM ESPERA MIENTRAS DA 2000 VUELTAS
6010 FOR I=1 TO 2000:NEXT I
6020 RETURN
8000 END
```

99. JUEGOS REUNIDOS (25.3)

Tienes un programa con tres juegos. El juego número uno es de marcianitos, el número dos de cartas y el tres es el dominó. Como eres algo olvidadizo no has puesto la forma de ir directamente a cada juego. ¿Cómo harías el inicio del programa para poder ir al juego deseado?

Si los juegos tienen sus comienzos en las líneas 2000, 4000 y 6000, respectivamente, será muy fácil empezar este programa de juegos con un módulo que sea como la subrutina 5000 del ejercicio anterior; de manera que diga cuáles son los juegos y los números que les corresponden y luego, cuando se elija número, si éste no vale lo diga y pida otro:

```
   10 REM JUEGOS REUNIDOS
(1)20 (BORRAR LA PANTALLA)
(2)30 PRINT TAB(12);"JUEGOS REUNIDOS"
(2)40 PRINT TAB(12);"***************"
   50 PRINT:PRINT
   60 PRINT TAB(3);"MARCIANITOS.............
      ........1":PRINT
   70 PRINT TAB(3);"CARTAS..................
      ........2":PRINT
   80 PRINT TAB(3);"DOMINO..................
      ........3":PRINT
   90 PRINT:PRINT "ELIGE JUEGO CON UN NUMERO
      "
   100 INPUT N
(3)110 ON N GOTO 2000,4000,6000:REM SE HA EL
       EGIDO
   120 PRINT "ESE NO VALE. !LISTO!"
   130 GOTO 100:REM SE REPITE
```

---

(1) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT "(SHIFT) (CLR HOME)".

(2) Si tu ordenador es un Spectrum, sustituye el número 12 por un 8.

(3) Si tu ordenador es un Spectrum, sustituye la línea 110 por ésta:

```
110 GOTO 2000*N
```

100. CALCULADORA (25.4)

Prepara un programa en el que el ordenador te pregunte dos números; a continuación, si quieres sumarlos, restarlos, multiplicarlos o dividirlos; y, según tu contestación, te presente el resultado.

Podemos poner en la pantalla un menú con las operaciones, para elegir una. El programa podría ser así:

```
   10 REM CALCULADORA
(1)20 (BORRAR LA PANTALLA)
   30 INPUT "DAME DOS NUMEROS";N1,N2
   40 PRINT:PRINT:PRINT
   50 PRINT "SEGUN EL NUMERO QUE ELIJAS, HAR
      E CON ELLOS...":PRINT
   60 PRINT "LA SUMA.........1"
   70 PRINT "LA RESTA........2"
   80 PRINT "EL PRODUCTO.....3"
   90 PRINT "LA DIVISION.....4"
   100 INPUT "QUE HAGO";H
(2)110 ON H GOTO 1000,2000,3000,4000
   120 REM SI QUIERES PONER ALGUN COMENTARIO
        PARA LOS "GRACIOSOS", HAZLO AQUI
   1000 REM SUMA
   1010 PRINT "LA SUMA DE";N1;" Y";N2;" VALE
         ";N1+N2
   1020 GOTO 5000
   2000 REM RESTA
   2010 PRINT N1;" MENOS";N2;" VALE";N1-N2
   2020 GOTO 5000
   3000 REM PRODUCTO
   3010 PRINT N1;" MULTIPLICADO POR";N2;" VA
         LE";N1*N2
   3020 GOTO 5000
   4000 REM DIVISION
   4010 PRINT N1;" DIVIDIDO ENTRE";N2;" VALE
         ";N1/N2
   5000 END
```

(1) Si tu ordenador es un Spectrum, pon CLS. Si es Commodore, pon PRINT "[SHIFT] [CLR HOME]".

(2) Si tu ordenador es un Spectrum, sustituye la línea 110 por ésta:

```
110 GOTO 1000*H
```

De los mismos autores:

- PROGRAMAS COMENTADOS DE BASIC BASICO
- BASIC JUNIOR. INICIACION A LA PROGRAMACION
- PROGRAMAS EXPLICADOS DE BASIC JUNIOR

También en esta colección:

- EL BASIC del Spectrum de A. Blanco y B. Compostela